Optimal control of Navier-Stokes equations using Lagrange-Galerkin methods

Gilles Fourestey, Marwan Moubachir

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Optimal control of Navier-Stokes equations using Lagrange-Galerkin methods

Gilles Fourestey — Marwan Moubachir

N° 4609

Octobre 2002

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4609--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Optimal control of Navier-Stokes equations using Lagrange-Galerkin methods

Gilles Fourestey*, Marwan Moubachir†

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet CAIMAN

Rapport de recherche n° 4609 — Octobre 2002 — 88 pages

Abstract: In this report, we are investigating the numerical approximation of an optimal control problem involving the evolution of a newtonian viscous incompressible fluid described by the Navier-Stokes equations. This PDE system is discretized using a low order finite element in space coupled with a Lagrange-Galerkin scheme for the nonlinear advection operator. We introduce a full discrete linearized scheme that is used to compute the gradient of a given cost function by ensuring its consistency. Using gradient based optimization algorithms, we are able to deal with two fluid flow control problems :

1. Drag reduction around a moving cylinder,
2. an identification of far-field velocity using the knowledge of the fluid load on a rectangular bluff body, for both fixed and moving configuration.

Key-words: Navier-Stokes equations, ALE formulation, optimal control, Lagrange-Galerkin

* INRIA-CAIMAN/ENPC-CERMICS, email: Gilles.Fourestey@inria.fr
† Laboratoire Central des Ponts et Chaussées, email: Marwan.Moubachir@lcpc.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Contrôle optimal des équations de Navier-Stokes par utilisation de schémas de type Lagrange-Galerkin

Résumé : Dans ce rapport, nous nous intéressons à un problème de contrôle optimal pour un système fluide en écoulement incompressible décrit par les équations de Navier-Stokes. Ce système d'EDP est discrétisé en utilisant une méthode d'éléments finis de bas degré couplée à une méthode de Lagrange-Galerkin pour le traitement de l'opérateur de convection non-linéaire. Nous décrivons un schéma linéarisé totalement discret afin de calculer de façon consistante le gradient d'une fonctionnelle de coût liée à la physique du problème. Nous traitons, alors, deux applications de contrôle en nous appuyant sur des algorithmes d'optimisation fondés sur le calcul du gradient :

1. Réduction de la trainée autour d'un solide cylindrique en rotation,
2. un problème d'identification de conditions aux limites en amont à partir de la connaissance des efforts fluides sur un profil rectangulaire, en configuration fixe ou mobile.

Mots-clés : équations de Navier-Stokes, formulation ALE, contrôle optimal, méthode de Lagrange-Galerkin

Contents

1 Introduction **7**

2 Mathematical setting **7**
2.1 Continuous cost function gradient 8
2.2 Gradient based optimization strategy 10
2.2.1 Conjugate gradient method 10
2.2.2 Quasi-Newton algorithms 11

3 The Lagrange-Galerkin Scheme for the Navier-Stokes equations **13**
3.1 Time discretisation 13
3.1.1 Introduction 13
3.1.2 Time semi-discrete first order characteristics scheme 13
3.1.3 Time semi-discrete second order characteristics scheme 14
3.2 Spatial approximation 15
3.2.1 Introduction 15
3.2.2 The P1-bubble/P1 space 16
3.3 Spatial approximation of the characteristic curve 17
3.4 Quadrature rules 18
3.5 Extension to the case of a moving domain 19
3.5.1 ALE formulation 19
3.5.2 Discretization 20

4 Discrete linearization and discrete cost function gradient **21**
4.1 Linearized discrete system 22
4.1.1 Linearized simple backtracking 22
4.1.2 Linearized subcycling backtracking 23
4.2 Linearized adaptative backtracking 24
4.2.1 2D case 25
4.2.2 3D case 27
4.3 Implementation using barycentric coordinates 27
4.3.1 Projection using barycentric coordinates 28
4.3.2 Derivative of the projection using barycentric coordinates 31
4.3.3 linearization of the discrete scheme inside NSI3 32
4.4 Discrete linearized system 34

5 Implementation **36**
5.1 Nonlinear and linearized Navier-Stokes solvers 36
5.1.1 NSI3 36
5.1.2 LGNSI2FS 37
5.2 Optimization routines 37
5.3 Parallel direct mode strategy 38

6 Discrete gradient computation **39**
6.1 Case of Poiseuille flow . . . 39
6.1.1 Flow tracking control problem . . . 39
6.1.2 Derivative checking : the intersection algorithm case . . . 40
6.1.3 Simple control test case . . . 42
6.1.4 derivative checking : case of the subcycling backtracking procedure . . 44
6.2 The driven cavity . . . 45
6.2.1 Definition of the problem . . . 45
6.2.2 Non-differentiability illustration for the adaptative backtracking scheme 45
6.2.3 Discrete gradient consistency using the subcycling backtracking algorithm . . . 46
6.2.4 Influence of the splitting parameter . . . 51
6.2.5 Second order LG scheme and integration rule . . . 54

7 Drag reduction around a rotating cylinder **56**
7.1 Control problem setting . . . 56
7.2 Case of a single harmonic angular velocity . . . 57
7.2.1 Continuous optimal control . . . 57
7.2.2 Numerical experiments . . . 58
7.3 Case of several harmonics . . . 60

8 Identification of far-field boundary conditions from fluid loads on bluff bodies **61**
8.1 Problem settings . . . 62
8.2 The rectangular cylinder . . . 64
8.3 Harmonic perturbation of the inflow velocity . . . 65
8.4 Synthetic load . . . 66
8.5 Arbitrary harmonic loads . . . 67

9 Identification of far-field boundary conditions from fluid loads on moving bluff bodies **69**
9.1 Problem settings . . . 69
9.2 Mesh movement algorithm . . . 70
9.3 Discrete gradient consistency . . . 73
9.4 Harmonic perturbation of the inflow velocity . . . 77

10 Conclusion **83**

List of Figures

1 2D projection along a given vector 25
2 Projection on the edge of a triangle 26
3 3D projection 27
4 Projection on the edge of a tetrahedron 28
5 Optimization tool structure 37
6 Parallel strategy for the gradient computation 38
7 Poiseuille Flow domain 41
8 First order convergence of the finite difference gradient for the Poiseuille Flow 41
9 Optimization process for the Poiseuille flow. 43
10 Tracking cost function decrease for the Poiseuille flow. 43
11 First and second order subcycling backtrackings scheme gradient checking with the theoretical slope for the Poiseuille flow 44
12 First order Lagrange-Galerkin scheme applied to the driven cavity flow - Gradient consistency checking with NSI3 46
13 First order and second order Lagrange-Galerkin scheme applied to the driven cavity flow - Gradient checking 47
14 l_2 norm of the driven cavity at steady state 48
15 l_2 norm of the linearized driven cavity at steady state 49
16 Splitting parameter efficiency for the second order LG scheme with $\Delta t = 0.2$ 52
17 Splitting parameter efficiency for the second order LG scheme with $\Delta t = 0.05$ 53
18 Computational mesh around the cylinder 59
19 3D contour plot of the cost function $j(\rho, S_e)$ 60
20 2D contour plot of the cost function $j(\rho, S_e)$ 61
21 Drag evolution for different control parameters 62
22 Lift evolution for different control parameters 63
23 Vortex shedding in the uncontrolled case 63
24 Vortex shedding in the optimal controlled case 63
25 Optimization parameters evolution 64
26 Optimal Drag time evolution - 3 harmonics case 65
27 Mean drag evolution- 3 harmonics case 66
28 Cost function gradient decrease - 3 harmonics case 67
29 Lift and drag for the R4 rectangular cylinder 68
30 Pressure profile for the R4 rectangular cylinder 69
31 Control parameters during optimization steps 70
32 Drag history - initial guess and final control 71
33 Gradient decrease during optimization steps 72
34 Lift history - initial guess and final control 73
35 Control parameters during optimization steps 74
36 Cost function decrease during optimization steps 75
37 Gradient decrease during optimization steps 76
38 Mesh vertices position at t = 0s (left) and $t = 16s$ (right) 77

39 First and second order Lagrange-Galerkin scheme applied to the moving R4 profile - Gradient test 78
40 Amplitude control parameter during the optimization steps 79
41 Frequency control parameter during the optimization steps 80
42 Cost function behaviour during the optimization steps 81
43 Gradient of the cost function behaviour during the optimization steps 82

1 Introduction

In the last two decades, the number of studies concerning optimal control problem for fluid mechanics has quickly increased. On the theoretical point of view, significant advances have been realized in the derivation of optimality systems [1, 19] or in the controllability properties investigation [11, 30, 14]. With the increasing power of computing capabilities, numbers of numerical studies have been conducted [32, 4, 24, 26, 29]. Most of these works involve finite difference schemes for the time discretization of the Navier-Stokes system. In this article, we propose to discretize the nonlinear-advection operator involved in the Navier-Stokes system using the characteristic method introduced by [3]. This method is based on the computation of characteristic paths in order to approximate the material time derivative. The use of this algorithm [9] introduces some technical difficulties addressed in this paper. Once the discrete linearized system is established, its solution enters the computation of cost function gradients involved in the optimization loop which minimizes a given objective. Two different applications are addressed:

- One concerning the drag reduction around a rotating circular cylindrical body using its angular velocity. This problem has been already treated by several authors [26, 29]. Our goal is mainly to validate our approach by analysing its ability of dealing with non-trivial control problem benchmarks.
- We also deal with an identification problem that may enter an aeroelastic stability tool described in [34]. It consists in trying to identify far-field boundary conditions from the knowledge of fluid loads time history on a given bluff body in fixed or prescribed moving configuration. The final application should be the aeroelastic stability analysis of civil engineering super-structures [39, 2, 12].

2 Mathematical setting

Let us consider a fluid inside a fixed domain Ω with boundary $\partial\Omega = \Gamma_c \cup \Gamma_s \cup \Gamma_{in} \cup \Gamma_{out}$. It is discribed by its velocity and its pressure (u, p) solution of the following Navier-Stokes system,

$$\left\{\begin{array}{ll} \partial_t u + \nabla u \cdot u - \nu\Delta u + \nabla p = 0, & \Omega \\ \mathrm{div}(u) = 0, & \Omega \\ u = g, & \Gamma_c \\ u = 0, & \Gamma_s \\ \sigma(u,p)\cdot n = 0, & \Gamma_{out} \\ u = u_\infty, & \Gamma_{in} \\ u(t=0) = u_0, & \Omega \end{array}\right. \tag{1}$$

where ν stands for the kinematic viscosity and u_∞ is the farfield velocity. The quantity

$$\sigma(u,p) = -p\,\mathrm{I} + \nu(\nabla u + \nabla^T u)$$

stands for the fluid stress tensor inside Ω, with $(\nabla u)_{ij} = \partial_j u_i = u_{i,j}$.
The aim of this paper is to control the behaviour of the fluid trough the control g on Γ_c, by minimizing a given cost function J depending on the control g and on the state solution $(u(g), p(g))$. This cost function may represent a given objective related to specific characteristic features of the fluid flow (e.g the drag on a given surface, the vorticity level, the distance to a given target). Hence we are interested in solving the following problem,

$$\min_{g \in \mathcal{U}_{ad}} J(u(g), p(g)) \tag{2}$$

where $(u(g), p(g))$ satisfies the Navier-Stokes system (1) associated to the control $g \in \mathcal{U}_{ad}$. In the applications, we shall deal with particular cost functions of the following type,

- The cost to overcome the drag on the solid interface or to track a given fluid load target,

$$J(u,p) = \int_0^T \left| \left(\int_{\Gamma_s} \sigma(u,p) \cdot n \, d\Gamma \right) \cdot e_1 - F_d(t) \right|^2 dt \tag{3}$$

- The mean square value to a given target,

$$J(u,p) = \int_0^T \int_{\Gamma_{out}} (u - u_d)^2 \, d\Omega \, dt \tag{4}$$

Dirichlet boundary control for the Navier-Stokes system has been addressed on the theoretical level in [19, 25] in the 2D case. This is a non trivial problem, especially concerning the regularity of the boundary control [18]. We shall not address such an issue in this paper, since we will use regular classes of controls in the applications. Our main interest here, is to describe how the continuous gradient of the cost function J can be obtained using the solution of a linearized fluid system. In a second step, we shall describe a continuous optimization algorithm based on the continuous gradient ∇J.

2.1 Continuous cost function gradient

Let us consider the control variable g as the element of an Hilbert space $\mathcal{U}_c$. Basic continuous feasible algorithms for minimizing the functional $j(g) \stackrel{\text{def}}{=} J(u(g), p(g))$ are based on the evaluation of the functional gradient against arbitrary direction inside $\mathcal{U}_c$. Through the paper, we shall use the following result,

Theorem 1 *The functional $j(g)$ is Gâteaux differentiable with respect to $g \in \mathcal{U}_c$ in every direction $\delta g \in \mathcal{U}_c$ and its derivative is given by the following expression,*

$$\langle j'(g), \delta g \rangle_{\mathcal{U}_c} = \langle \partial_{(u,p)} J(u(g), p(g)), (z(\delta g), q(\delta g)) \rangle, \quad \forall \delta g \in \mathcal{U}_c \tag{5}$$

where $(z(\delta g), q(\delta g)) \stackrel{\text{def}}{=} \left(\frac{\mathrm{D}}{\mathrm{D}\, g}(u,p)\right)(g)\cdot\delta g$ *stands for the linearized state which is solution of the linearized Navier-Stokes system,*

$$\left\{\begin{array}{ll} \partial_t z + \nabla u(g)\cdot z + \nabla z\cdot u(g) - \nu\Delta z + \nabla q = 0, & \Omega \\ \mathrm{div}(z) = 0, & \Omega \\ z = \delta g, & \Gamma_c \\ z = 0, & \Gamma_s \\ \sigma(z,q)\cdot n = 0, & \Gamma_{out} \\ z = 0, & \Gamma_{in} \\ z(t=0) = 0, & \Omega \end{array}\right. \tag{6}$$

The linearized system is a forward in time linear advection-diffusion-reaction problem. It needs the knowledge of the original fluid flow $u(g)$ which is specific to non-linear control problems. If we want to identify the functional gradient, we need to evaluate the linearized flow for every direction $\delta g \in \mathcal{U}_c$. Concerning the discretization issue, this can lead very quickly to expensive computations. For this reason, people usually use an adjoint continuous formulation in order to use a control space of arbitrary size, as described in the following theorem,

Theorem 2 *The functional* $j(g) = \dfrac{1}{2}\displaystyle\int_0^T\int_{\Gamma_{out}} (u(g)-u_d)^2\, d\Omega\, dt$ *is Fréchet differentiable with respect to* $g\in\mathcal{U}_c$ *and its gradient is given by the following expression,*

$$\nabla j(g) = -\sigma(\varphi,\pi)\cdot n|_{\Gamma_c\times(0,T)} \tag{7}$$

where (φ,π) *stands for the adjoint state which is solution of the adjoint Navier-Stokes system,*

$$\left\{\begin{array}{ll} -\partial_t\varphi - \nabla\varphi\cdot u(g) + \nabla^T u(g)\cdot\varphi - \nu\Delta\varphi + \nabla\pi = 0, & \Omega \\ \mathrm{div}(\varphi) = 0, & \Omega \\ \varphi = 0, & \Gamma_c \\ \varphi = 0, & \Gamma_s \\ \sigma(\varphi,\pi)\cdot n = u(g) - u_d, & \Gamma_{out} \\ \varphi = 0, & \Gamma_{in} \\ \varphi(t=T) = 0, & \Omega \end{array}\right. \tag{8}$$

The adjoint problem is a backward linear advection-diffusion-reaction system. As for the linearized system, it needs the knowledge of the primal state $u(g)$ in order to be solved. Contrary to the linearized state which can be advanced in time with the solution $u(g)$ of the state problem, the adjoint system has a final time condition, which means that it has to be solved once the Navier-Stokes system has been solved. Several authors have been using this formulation in order to solve complex fluid flow control problems [26, 42, 8]. In this paper, we will not use such an approach and prefer a direct approach involving the linearized system.

2.2 Gradient based optimization strategy

At this stage, several optimization strategies can be considered, we have choosen to investigate the use of two iterative optimization procedures that may need only the computation of the cost function gradient at each minimization step. The first one is refered as the conjuguate gradient algorithm in the context of nonlinear - nonconvex optimization problems and is usually seen as a first order optimization algorithm. The second one is refered as the BFGS method and belongs to the second order quasi-Newton optimization algorithms class. We shall describe both strategies in the next section in an abstract setting.

2.2.1 Conjugate gradient method

Let us consider the following abstract minimization problem,

$$\left\{ \begin{array}{l} u \in H, \\ j(u) \leq j(v), \quad \forall\, v \in H \end{array} \right. \tag{9}$$

where H stands for an Hilbert space and $j : H \rightarrow \mathbb{R}$ is a differentiable functional on H. The Conjuguate Gradient algorithm applied to problem (9) can be described in the following manner,

- Step 0

  $$\text{Choose } u^0 \in H$$

  $$\left\{ \begin{array}{ll} \text{Solve for } g^0 \in H, & \text{such that} \\ (g^0, v)_H = {}_{H^*} < j'(u^0), v >_H, & \forall\, v \in H \end{array} \right.$$

  We set

  $$w^0 = g^0$$

- Step $n+1$

  For $n \geq 0$, we assume that (u^n, g^n, w^n) is at our disposal, and we seek for $(u^{n+1}, g^{n+1}, w^{n+1})$ thanks to the following steps :

  1. Line search: Compute ρ_n such that,

     $$(LS) \left\{ \begin{array}{ll} \text{Solve for } \rho_n \in \mathbb{R}, & \text{such that} \\ j(u^n - \rho_n w^n) \leq j(u^n - \rho w^n), & \forall\, \rho \in \mathbb{R} \end{array} \right.$$

     We set

     $$u^{n+1} = u^n - \rho_n w^n,$$

2. Gradient computation :

$$\begin{cases} \text{Solve for } g^{n+1} \in H, & \text{such that} \\ (g^{n+1}, v)_H = {}_{H^*} < j'(u^{n+1}), v >_H, & \forall\, v \in H \end{cases}$$

3. Convergence test:
   If $\frac{\|g^{n+1}\|}{\|g^0\|} \leq \epsilon$, the solution is given by

$$u = u^{n+1}$$

   unless, we compute one of the following quantities,
   a)

$$\gamma_n = \frac{\|g^{n+1}\|^2}{\|g^n\|^2}, \text{ Fletcher-Reeves}$$

   b)

$$\gamma_n = \frac{(g^{n+1}, g^{n+1} - g^n)}{\|g^n\|^2}, \text{ Polack-Ribière}$$

   Then, we set

$$w^{n+1} = g^{n+1} + \gamma_n w^n$$

4. Iteration loop : we set $n := n + 1$, we come back to step $n + 1$.

The investigation of the convergence of the Conjugate gradient algorithm was performed on the infinite dimensional level for quadratic convex functionals. We refer to [27] for a review of such results. These results cannot be applied in our case, since we deal with a non-convex, non-linear functional. Still, on the numerical point of view the CG method has shown great convergence features in many applications, and particularly for optimal control of fluid models [23, 28].

2.2.2 Quasi-Newton algorithms

For very large optimization problems involving very flat functionals, it has been shown that the CG method can converge very slowly. In order to circumvent this issue, it happens to be useful to use quasi-Newton methods that may show second order convergence properties near the optimum. One possible choice is the use of the BFGS method as described by Liu-Nocedal in [31]. This method is based on the approximation of the cost function Hessian inverse using a rank one perturbation of identity deduced from the gradient knowledge. This method was successfully applied to optimal control of Navier-Stokes equations in [4],[26].

We consider the abstract minimization problem (2). For $(u,v) \in H \times H$, we consider the element $u \otimes v \in \mathcal{L}(H)$ defined as follows,

$$(u \otimes v)(w) =< w, v >_H u, \quad \forall w \in H$$

Hence the BFGS algorithm reads as follows,

- Step 0 : Choose $u^0 \in H$,$H^0 \in \mathcal{L}(H)$

$$\left\{ \begin{array}{ll} \text{Solve for } g^0 \in H, & \text{such that} \\ (g^0, v)_H = {}_{H^*} < j'(u^0), v >_H, & \forall\, v \in H \end{array} \right.$$

- Step $n+1$

  For $n \geq 0$, we assume that (u^n, g^n, H^n) is at our disposal, and we seek for $(u^{n+1}, g^{n+1}, H^{n+1})$ thanks to the following steps :
  We set $d^n = -H^n \cdot g^n$,

  1. Line search: Compute ρ_n such that,

$$(LS) \left\{ \begin{array}{ll} \text{Solve for } \rho_n \in \mathbb{R}, & \text{such that} \\ j(u^n + \rho_n d^n) \leq j(u^n + \rho d^n), & \forall\, \rho \in \mathbb{R} \end{array} \right.$$

     We set

$$u^{n+1} = u^n + \rho_n w^n,$$

  2. Gradient computation :

$$\left\{ \begin{array}{ll} \text{Solve for } g^{n+1} \in H, & \text{such that} \\ (g^{n+1}, v)_H = {}_{H^*} < j'(u^{n+1}), v >_H, & \forall\, v \in H \end{array} \right.$$

  3. Convergence test : If $\frac{\|g^{n+1}\|}{\|g^0\|} \leq \epsilon$, the solution is given by

$$u = u^{n+1}$$

     unless perform step 4.

  4. Hessian update :
     We set

$$s^n = u^{n+1} - u^n$$

     and

$$y^n = g^{n+1} - g^n$$

     Then we compute

$$H^{n+1} = H^n + \frac{d^n \otimes d^n}{< d^n, s^n >_H} - \frac{H^n \cdot s^n \otimes H^n \cdot s^n}{< H^n \cdot s^n, s^n >_H}$$

  5. Iteration loop : we set $n := n+1$, we come back to step $n+1$.

3 The Lagrange-Galerkin Scheme for the Navier-Stokes equations

In this paragraph, we shall describe our strategy for the finite dimensional approximation of the solution of the Navier-Stokes system, based on a Lagrange-Galerkin scheme.

3.1 Time discretisation

3.1.1 Introduction

The Lagrange-Galerkin scheme was first introduced by Benqué [3] and is made of the combination of the characteristics method with a standard finite-element formulation. The main idea of this method lies in the fact that the operator $\frac{\partial}{\partial t} + u.\nabla$ may be turned into a total derivative $\frac{d}{dt}$ using a Lagrangian formulation. However, this Lagrangian formulation is only valid along the characteristics curves of the particule. This curves are described by the following equations:

$$\left\{ \begin{array}{ll} \dfrac{d}{d\tau}\chi(\tau;t,x) = u(\tau,\chi(\tau;t,x)), \tau \in [0,t] & \\ \chi(\tau = t;t,x) = x & x \in \Omega \end{array} \right. \tag{10}$$

or equivalently :

$$\chi(\tau;t,x) = x + \int_0^t u(\tau,\chi(\tau;t,x))d\tau \tag{11}$$

Hence, the characteristics method consists in performing the following steps :

1. Define an approximation scheme for the Cauchy problem (10). Solving this system leads to an approximation of the characteristic curves.
2. Define a time approximation scheme for the total derivative operator using the approximate characteristic curves.
3. Solve the resulting generalized Stokes system.

Remark 1 *The non-linear part of the Navier-Stokes system is hidden in the Cauchy problem (10), and the generalized Stokes system is linear.*

3.1.2 Time semi-discrete first order characteristics scheme

We set

$$\begin{array}{rl} (u^n,p^n,g^n)(x) = (u,p,g)(x,t_n), & x \in \bar{\Omega} \\ t_n = t_0 + n\Delta t, \quad t_0 = 0,\ t_N = T & \end{array}$$

We consider the characteristic curve associated to the flow field u as the solution of the Cauchy problem (10). Hence, using the chain rule, we get the following expression for the time derivative of the flow field u along a characteristic curve ,

$$\frac{d}{d\tau}u(\tau,\chi(\tau;t,x))|_\tau = \partial_\tau u(\tau,\chi(\tau;t,x))|_\tau + \nabla u(\tau,\chi(\tau;t,x))\cdot u(\tau,\chi(\tau;t,x))$$

For $t = t_{n+1}, t_n \leq \tau \leq t_{n+1}$, we define the first-order approximation of the total time derivative (Backward Euler scheme),

$$\frac{d}{d\tau}u(\tau,\chi(\tau;t,x))|_{\tau=t_{n+1}} = \frac{u(t_{n+1},x) - u(t_n,\chi(t_n;t_{n+1},x))}{\Delta t}$$

The characteristic foot$\chi(t_n;t_{n+1},x)$ is computed from (11) using the following linear discrete interpolation :

$$\chi(t_n;t_{n+1},x) = x - \Delta t u^n(x) \tag{12}$$

Setting $\chi^n(x) = \chi(t_n;t_{n+1},x)$, we finally obtain the time semi-discrete first order Navier-Stokes system (see [16] for the full proof) :

$$\begin{cases} \frac{1}{\Delta t}u^{n+1} - \nu\Delta u^{n+1} + \nabla p^{n+1} = \frac{1}{\Delta t}u^n\circ\chi^n, & \Omega \\ \mathrm{div}(u^{n+1}) = 0, & \Omega \\ u^{n+1} = g^{n+1}, & \Gamma_{cyl} \\ \sigma(u^{n+1},p^{n+1})\cdot n = 0, & \Gamma_{out} \\ u^{n+1} = u_\infty^{n+1}, & \Gamma_{in} \end{cases} \tag{13}$$

with $u^{t=0} = u_0$. One of the greatest advantage of this formulation is that large time steps may be used in conjunction with the Lagrange-Galerkin method. Unlike its Eulerian counterpart, the Lagrangian formulation is not restricted, at least theoretically, by any CFL. However, working with large time steps may induce large numerical dissipation. An increase of the order of the time dicretization scheme can eventually reduce this overdissipation.

3.1.3 Time semi-discrete second order characteristics scheme

The second order Lagrange-Galerkin method was introduced by Boukir et al [6] and is an extension of the first order scheme. As before, let us define the total derivative :

$$\frac{d}{d\tau}u(\tau,\chi(\tau;t,x))|_{\tau=t_{n+1}} = \frac{3u(t_{n+1},x) - 4u(t_n,\chi(t_n;t_{n+1},x)) + u(t_{n-1},\chi(t_{n-1};t_{n+1},x))}{2\Delta t}$$

Now we need to compute two sets of characteristics paths :

$$\begin{cases} \chi_1^n = \chi(t_n;t_{n+1},x) & = x - \Delta t(2u^n - u^{n-1})(x) \\ \chi_2^n = \chi(t_{n-1};t_{n+1},x) & = x - 2\Delta t(2u^n - u^{n-1})(x) \end{cases}$$

Then, we get the time semi-discrete second order Navier-Stokes system [6] [16],

$$\begin{cases} \frac{3}{2\Delta t}u^{n+1} - \nu\Delta u^{n+1} + \nabla p^{n+1} = \frac{1}{\Delta t}\left[2u^n \circ \chi_1^n - \frac{1}{2}u^{n-1} \circ \chi_2^n\right], & \Omega \\ \mathrm{div}(u^{n+1}) = 0, & \Omega \\ u^{n+1} = g^{n+1}, & \Gamma_{cyl} \\ \sigma(u^{n+1}, p^{n+1}) \cdot n = 0, & \Gamma_{out} \\ u^{n+1} = u_\infty^{n+1}, & \Gamma_{in} \end{cases} \tag{14}$$

We may note that the general form of the second order scheme is similar to the one used for the first order scheme. Apart form the coefficient of the mass matrix and the two sets of characteristics path needed to perform the computation of the right hand side for the second order, both Lagrange Galerkin formulations may be solved using the same procedure. Nevertheless, the right hand side computation for the second order is much more cpu time consuming than the first order because instead of tracking the particle using one velocity field over a time period of Δt, the second order will require another tracking over a time period of $2\Delta t$. This may substantially slow down the characteristics path construction procedure. In case the tracking process is performed with poor numerical accuracy, some numerical error may impair the whole flow computation. This issue will be discussed in the next section.

3.2 Spatial approximation

3.2.1 Introduction

Here we describe several points that must be handled carefully in order to compute a reasonable numerical approximation of the Navier-Stokes system.
When the viscosity constant ϵ is small, the Navier-Sokes equations are strongly non-linear. This imply a severe restriction over the time step, but the use of the characteristics method may overcome this difficulty, as long as the characteristics path is computed correctly. Then, the incompressible constraint confers the Navier-Stokes equations with a strong saddle point character. This implies the velocity and the pressure functional spaces to be chosen with care. In order to correctly cope with this difficult, the LBB condition [20] [22] must be satisfied. If X_h and M_h are the discretized spaces for the velocity and the pressure respectively, then, for all $v \in X_h$ and $p \in M_h$:

$$\inf_{p \in M_h \setminus Q_h} \sup_{v \in X_h} \frac{|(\nabla . v, p)|}{||v||_{X_h}||p||_{M_h}} \geq \gamma > 0 \tag{15}$$

with $Q_h = \{q \in M_h | (q, \nabla . v) = 0 \quad \forall v \in X_h\}$ and γ is a constant independent of h. A set of spaces satisfying this condition may be found in [40] and [20]. Among them, we shall use the popular P1-bubble/P1 subspace, which is the smallest subspace satisfying the LBB condition.

3.2.2 The P1-bubble/P1 space

Let us suppose that $\Omega \in \mathbb{R}^d$ is a polyhedral set, and let us consider a $d-$ simplex partition $\mathcal{T}_h$ of Ω. We suppose that $(\mathcal{T}_h)_{h\geq 0}$ satisfies standard hypothesis in order to define a finite element family[22].
We set,

$$P_h^1 = \left\{\phi : \Omega \to \mathbb{R} \ / \ \phi|_K \in \mathbb{P}^1, \ \forall \ K \in \mathcal{T}_h\right\}$$

For $k \in \mathcal{T}_h$, we consider $\{\lambda_i\}_{1\leq i\leq d+1}$, the barycentric coordinates with respect to the $d-$ simplex K. We introduce the bubble function

$$b^K = \prod_{i=1}^{d+1} \lambda_i$$

and we set

$$P_K^b = vect\{b^K\}$$

We define an approximation space pair based on the above space,

$$X^h = \left\{v \in (\mathcal{C}^0(\overline{\Omega}))^d, v|_K \in \left(\mathbb{P}^1 \oplus \mathbb{P}^b(K)\right)^d, \ \forall \ K \in \mathcal{T}_h\right\} \tag{16}$$

$$M_h = P_h^1 \bigcap \mathcal{C}^0(\overline{\Omega}) \tag{17}$$

Remark 2

- *The nodes of elements of M_h are defined as the vertex of the simplex,*
- *the nodes of elements of X_h are defined as vertex and isobarycentre of the simplex,*
- *(X_h, M_h) satisfies the LBB condition [22] [20] [16].*

We introduce the following functional spaces,

$$V = \left\{v \in (H^1(\Omega))^d, v = 0 \text{ on } \Gamma_{in} \cup \Gamma_s, \ v \cdot n = 0, \ \Gamma_{lat}\right\}$$

$$Q = L^2(\Omega)$$

We set $V_h = X_h^3 \cap V$ and $Q_h = M_h \cap Q$, and we look for $(u_h^{n+1}, p_h^{n+1}) \in \ X_h^3 \times Q^h$ solution of the following system :

$$\left\{\begin{array}{ll} \frac{1}{\Delta t}(u_h^{n+1}, v_h) + \nu(\nabla u_h^{n+1}, \nabla v_h) - (p_h^{n+1}, \operatorname{div} v_h) = \frac{1}{\Delta t}(u_h^n \circ \chi_h^n, v_h), & \forall \ v_h \in V_h \\ (\operatorname{div} u_h^{n+1}, q_h) = 0, & \forall \ q_h \in Q_h \\ u_h^{n+1} = \Pi_h g^{n+1}, & \Gamma_{cyl} \\ \sigma(u_h^{n+1}, p_h^{n+1}) \cdot n = 0, & \Gamma_{out} \\ u_h^{n+1} = \Pi_h u_\infty^{n+1}, & \Gamma_{in} \end{array}\right. \tag{18}$$

for the first order scheme and

$$\left\{\begin{array}{ll} \dfrac{3}{2\Delta t}(u_h^{n+1}, v_h) \quad +\nu(\nabla u_h^{n+1}, \nabla v_h) - (p_h^{n+1}, \operatorname{div} v_h) & \\ \qquad = \dfrac{2}{\Delta t}(u_h^n \circ \chi_{1,h}^n, v_h) - \dfrac{1}{2\Delta t}(u_h^{n-1} \circ \chi_{2,h}^n, v_h) & \forall\ v_h \in V_h \\ (\operatorname{div} u_h^{n+1}, q_h) = 0, & \forall\ q_h \in Q_h \\ u_h^{n+1} = \Pi_h g^{n+1}, & \Gamma_{cyl} \\ \sigma(u_h^{n+1}, p_h^{n+1}) \cdot n = 0, & \Gamma_{out} \\ u_h^{n+1} = \Pi_h u_\infty^{n+1}, & \Gamma_{in} \end{array}\right. \tag{19}$$

where χ_h^n stands for an approximation of the characteristic foot coming from point $x \in \Omega$. Existence and uniqueness results of the solution of problem (18) and (19) are well known, whereas regularity results are insured by the LBB condition. The details of this approximation will be specified in the next section.

3.3 Spatial approximation of the characteristic curve

As stated before, severe restrictions on the CFL may be relaxed using the Lagrange-Galerkin scheme. However, when large time steps are used, computing the characteristics path efficiently becomes crucial. We shall now describe three backtracking schemes,

a) Simple backtracking :
This is the simplest algorithm for characteristics path approximation. The feet of the characteristics are computed using the following formulae, obtained directly from the characteristic path definition :

$$\chi_h^n(x) = x - \Delta t \cdot u_h^n(x),\ x \in \Omega$$

Here, the characteristic path is considered to be a straight line from the starting point x to its foot $\chi_h^n(x)$. This algorithm, though very simple provided we have a good particule tracking algorithm, performs badly when the time step is relatively large. In fact, when tracking the particles in a finite element mesh, one should not jump too many cells at once in order to ensure consistency. Hence, this algorithm should be used only when small time steps are used.

b) Subcycling backtracking :
Depending on the discrete velocity field $u_h^n(.)$, the last approximation may be more or less accurate, specially if the characteristic foot χ_h^n is far from its origin point x. Hence it may be convenient to perform a refinement of such an approximation :
The characteristic foot $\chi^n(.)$ is approximated by the extreme point of a polygonal curve of time length Δt with vertex $(\chi_0^n = x, \ldots, \chi_m^n = \chi^n(x))$ such that,

$$\chi_{i+1}^n = \chi_i^n - \Delta t_i \cdot u^n(\chi_i^n)) \text{ with } \sum_{i=1}^{m} \Delta t_i = \Delta t$$

The major drawback of the last scheme is that it requires, on the discrete level, to have an efficient element searching tool for the velocity field interpolation at each substep points χ_i^n. This may be even more CPU time consuming if we use 3D unstructured meshes. Whatsmore, the subdivision parameter is not mesh-dependent and may be easily oversized without necessarily increasing the accuracy level of $\chi^n(.)$.

c) Adaptative backtracking :
As stated previously, it may be useful to overcome the use of an element searching tool while using unstructured meshes. This may be done by using the following scheme as in [40] :
The characteristic foot $\chi^n(.)$ is approximated by the extreme point of a polygonal curve of time length Δt with vertex $(\chi_0^n = x, \ldots, \chi_m^n = \chi^n(x))$ where $x \in K_h^0$ such that,

1. $\chi_{i+1}^n = \chi_i^n - \Delta t_i \cdot u^n(\chi_i^n)$,
2. Δt_i is such that $\chi_i^n,\ i \in [1, m-1]$ belongs to the edge of the simplex of K_h^i and $\sum_{i=1}^{m} \Delta t_i = \Delta t$.

It follows that, in each element, the characteristic path is approximated by a straight line until the edge is reached. A complete description of these algorithm may be found in [16].

Remark 3

1. *As we may see in the sequel, the intermediate points χ_{i+1}^n can be viewed as the projection along the vector $u^n(\chi_i^n)$ on the edge of the simplex K_h^i.*
2. *It may be emphasized that the time partition depends on the geometry of the mesh and the velocity field $u_h^n(.)$.*
3. *Because of adaptativity, the number of sub-time steps is not exactly controlled.*

3.4 Quadrature rules

Computing the right hand side of the Lagrange-Galerkin system exactly is extremely expensive. Therefore, we shall use a Gauss quadrature formula. Introducing the family of Gauss points $(\alpha_i, \zeta_i)_{i \in I}$, we have

$$(u_h^n \circ \chi_h^n, v_h)_K = \sum_{i \in I} \alpha_i u_h^n(\chi_h^n(\zeta_i)) \cdot v_h(\zeta_i)$$

We can see that the characteristic path will be computed from each Gauss point. It follows that increasing the number of Gauss points in order to reach higher accuracy will result in an increase of the number of characteristics paths. This makes the characteristic feet location procedure even more expensive, especially when large time steps are used.

Remark 4 *Because the characteristics foot χ_h^n does not necessarily belong to the degree of freedom family of the discrete space X_h, the unconditional stability of the exact Lagrange-Galerkin scheme is lost. It has been proven that using a non-exact integration rule for the Lagrange-Galerkin method leads to a conditionnaly stable scheme, with non-linear CFL-like conditions (see [33] and [15] for more details). However, these instabilities may be reduced, though not entirely removed, by using a large number of Gauss points.*

3.5 Extension to the case of a moving domain

We may now change the problem configuration. Indeed we consider that the fluid domain is no more fixed, but can move with prescribed evolution. We note by Ω_t^f the moving domain at time $t > 0$. We endow the system with a sticking boundary condition at the fluid-solid interface,

$$u(x,t) = \dot{x}^s, \quad (x,t) \in \Gamma_t^s \times (0,T) \tag{20}$$

We may only deal in the sequel with rigid body displacement fields, i.e

$$x^s(x_0,t) = d(t) + Q(t)\,(x_0 - x_G), \quad \text{on } \Gamma_0^s \tag{21}$$

where,

$$Q(t) = Q_0 \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

stands for the rotation operator and x_G the center of mass of the solid. The couple (u,p) satisfies the Navier-Stokes equations in moving domain,

$$\begin{cases} \partial_t u + \nabla u \cdot u - \nu\Delta u + \nabla p = 0, & \Omega_t^f \\ \mathrm{div}(u) = 0, & \Omega_t^f \\ u = \dot{x}^s, & \Gamma_t^s \\ \sigma(u,p)\cdot n = 0, & \Gamma_{out} \\ u = u_\infty, & \Gamma_{in} \\ u(t=0) = u_0, & \Omega \end{cases} \tag{22}$$

3.5.1 ALE formulation

The previous formulation is not convenient in view of its approximation. Indeed, the temporal derivation part $\partial_t|_{x\in\,\Omega_t^f}$ is computed while fixing the spatial variable x in a time-dependent domain Ω_t^f. Then, in the neighbourhood of the moving boundary Γ_t^s, it may be difficult to define an approximation of such a derivative.
A solution consists in introducing a map that will transform the term $\partial_t|_{x\in\,\Omega_t^f}$ into a temporal derivative term at time $t \geq 0$, keeping the space variable fixed in a domain which does not dependent on the time variable t.

In order to work with a Lagrangian description of the solid dynamic, we choose an invariant moving boundary map along Γ_t^s. Hence, we introduce the following mapping family, for $0 \leq \tau < 0$,

$$\mathcal{A}_t^\tau : \begin{array}{ccc} \overline{\Omega}^f(\tau) & \longrightarrow & \overline{\Omega}^f(t) \\ \xi & \longmapsto & x = \mathcal{A}_t^\tau(\xi) \end{array}$$

such that

$$\mathcal{A}_t^\tau(\xi) \in \Gamma_t^s, \quad \forall \xi \in \Gamma_0^s$$

We set $x^f = \mathcal{A}_t^\tau(\xi)$ for $\xi \in \Omega^f(\tau)$ and $x^s \stackrel{\text{def}}{=} \mathcal{A}_t^\tau(\xi)$ for $\xi \in \Gamma^s(\tau)$. Then the ALE map can be described as follow,

$$\begin{array}{rcll} x^f(\xi,t) & = & \text{Ext}(x^s)(\xi,t), & \forall \xi \in \Omega^f(\tau), \\ x^s(\xi,t) & = & d(t) + Q(t)\,\xi, & \forall \xi \in \Gamma^s(\tau). \end{array}$$

Here, Ext represents any extension operator from $\Gamma^s(\tau)$ to $\Omega^f(\tau)$ preserving Γ_∞^f.
$w^\tau(x,t) = \frac{\partial \mathcal{A}_t^\tau(\xi)}{\partial t}\Big|_{\xi \in \Omega^f(\tau)}$ with $x = \mathcal{A}_t^\tau(\xi)$, stands for the ALE velocity.
In the sequel, we will not distinguish between a function defined in $\Omega^f(\tau)$ or $\Omega^f(t)$, keeping in mind that they are connected through the ALE map $(\mathcal{A}_t^\tau)_{0<\tau<t}$. Using the Chain Rule, we may state the following identity,

$$\frac{\partial u(x,t)}{\partial t}_{|\xi \in \Omega^f(\tau)} = \frac{\partial u(x,t)}{\partial t}_{|x \in \Omega^f(t)} + \nabla u(x,t) \cdot w^\tau(x,t), \text{for } x = \mathcal{A}_t^\tau(\xi), \ \forall \xi \in \Omega^f(\tau) \quad (23)$$

It allows us to obtain the strong ALE formulation of the fluid-structure coupled system,

$$\left\{ \begin{array}{ll} \frac{\partial u}{\partial t}_{|\xi \in \Omega^f(\tau)} + \nabla u \cdot (u - w^\tau) - \nu \Delta u + \nabla p = 0, & \Omega_t^f \\ \text{div}(u) = 0, & \Omega_t^f \\ u = \dot{x}^s, & \Gamma_t^s \\ \sigma(u,p) \cdot n = 0, & \Gamma_{out} \\ u = u_\infty, & \Gamma_{in} \\ u(t=0) = u_0, & \Omega \end{array} \right. \quad (24)$$

3.5.2 Discretization

The ALE formulation of the Navier-Stokes system in moving domain has a similar structure compared to the fixed domain case. The only difference is the convection velocity field which is shifted by the ALE velocity field. Hence the Lagrange Galerkin scheme can be applied and the characteristic curves are backtracked thanks to the field $u^n - w^n$. This leads to the following semi-discrete systems,

- First-order characteristic scheme :

$$\begin{cases} \frac{1}{\Delta t}u^{n+1} - \nu\Delta u^{n+1} + \nabla p^{n+1} = \frac{1}{\Delta t}u^n \circ \chi^n_{u-w}, & \Omega^{n+1} \\ \mathrm{div}(u^{n+1}) = 0, & \Omega \\ u^{n+1} = w^{n+1}, & \Gamma^s_{n+1} \\ \sigma(u^{n+1}, p^{n+1}) \cdot n = 0, & \Gamma_{out} \\ u^{n+1} = u^{n+1}_\infty, & \Gamma_{in} \end{cases} \tag{25}$$

  where $\chi^n_{u-w}(x) = \chi(t_n; t_{n+1}, x)$ stands for a characteristic foot obtained through the characteristic path $\chi(t_n; t_{n+1}, x)$ which is computed from (11) using the following formulae (linear discrete interpolation):

$$\chi(t_n; t_{n+1}, x) = x - \Delta t(u^n - w^n)(x)$$

- Second-order characteristic scheme :

$$\begin{cases} \frac{3}{2\Delta t}u^{n+1} - \nu\Delta u^{n+1} + \nabla p^{n+1} = \frac{1}{\Delta t}\left[2u^n \circ \chi^n_{u-w,1} - \frac{1}{2}u^{n-1} \circ \chi^n_{u-w,2}\right], & \Omega^{n+1} \\ \mathrm{div}(u^{n+1}) = 0, & \Omega \\ u^{n+1} = w^{n+1}, & \Gamma^s_{n+1} \\ \sigma(u^{n+1}, p^{n+1}) \cdot n = 0, & \Gamma_{out} \\ u^{n+1} = u^{n+1}_\infty, & \Gamma_{in} \end{cases} \tag{26}$$

  with the following set of characteristic feet,

$$\begin{cases} \chi^n_{u-w,1} = \chi(t_n; t_{n+1}, x) & = x - \Delta t(2(u^n - w^n) - (u^{n-1} - w^{n-1}))(x) \\ \chi^n_{u-w,2} = \chi(t_{n-1}; t_{n+1}, x) & = x - 2\Delta t(2(u^n - w^n) - (u^{n-1} - w^{n-1}))(x) \end{cases}$$

4 Discrete linearization and discrete cost function gradient

In order to perform our optimal control strategy, we need to compute the gradient of the cost function with respect to boundary velocity field. This may be done by using the derivative (z, q) of the state variable (u, p) with respect to $g \in \mathcal{U}_{ad}$. Concerning the numerical approximation of the optimization procedure, there exists mainly two strategies for computing the cost function gradient:

1. Define an approximation of the continuous linearized problem.
2. Define a linearization of the discretized state problem.

On the limit case $(h, \Delta t) \to 0$, both systems will converge to the continuous linearized system. Nevertheless, since we shall perform an optimization descent algorithm, both systems may furnish different descent directions, inducing different convergence properties. In the sequel, we may describe such an alternative. We will only deal with first-order characteristic linearized schemes, since the second-order case derives easily from the first-order case as shown before.

4.1 Linearized discrete system

The procedure which consists in,

1. discretizing the state problem,
2. then linearizing the discrete problem.

will produce what we call the linearized discrete problem (LDP). Let us describe this procedure for the Navier-Stokes problem. We set $(u_h^i(g), p_h^i(g))_{0 \leq i \leq N_t}$ the solution of the full-discrete Navier-Stokes system associated with the control variable g.
We would like to evaluate the following quantities,

$$(z_h^i(\delta g), q_h^i(\delta g)) = (\frac{Du_h^i(g)}{Dg} \cdot \delta g, \frac{Dq_h^i(g)}{Dg} \cdot \delta g)$$

For that purpose, the only difficulty is to differentiate the following term, involved in the right hand side of the generalized Stokes system,

$$\left(\frac{D(u_h^n \circ \chi_h^n(g))}{Dg}\right) \cdot \delta g = \frac{D(u_h^n)}{Dg}(\chi_h^n(g)) \cdot \delta g + \nabla u_h^n(\chi_h^n(g)) \cdot \frac{D(\chi_h^n(g))}{Du} \cdot \left(\frac{D(u_h^n)}{Dg}\right) \cdot \delta g \tag{27}$$

We recall that $z_h^n = \dfrac{D(u_h^n)}{Dg} \cdot \delta g$, then we get

$$\left(\frac{D(u_h^n \circ \chi_h^n(g))}{Dg}\right) \cdot \delta g = z_h^n(\delta g) \circ \chi_h^n(g) + \nabla u_h^n(\chi_h^n(g)) \cdot \left(\frac{(D\chi_h^n(g))}{Du}\right) \cdot z_h^n(\delta g) \tag{28}$$

This term involves the derivative of the characteristic foot with respect to the convection velocity field in the direction of the linearized velocity field. Hence depending on the characteristics backtracking procedure, we may get different derivative expressions. This issue will be investigated in the sequel.

4.1.1 Linearized simple backtracking

In this simple case, we recall that the characteristic foot $\chi(z)$ is approximated by a line,

$$\chi(z) = x - \Delta t \cdot z(x)$$

It follows easily that,

$$\frac{D\chi}{Dz} \cdot \delta z = -\Delta t \cdot \delta z(x)$$

Then we get

$$\left(\frac{D(u_h^n \circ \chi_h^n(g))}{Dg}\right) \cdot \delta g = z_h^n(\delta g) \circ \chi_h^n(g) - \Delta t \cdot \nabla u_h^n(\chi_h^n(g)) \cdot z_h^n(\delta g) \tag{29}$$

We finally need to solve the following linearized problem,

$$\begin{cases} \dfrac{1}{\Delta t}(z_h^{n+1}, v_h) + \nu\left(\nabla z_h^{n+1}, \nabla v_h\right) - (q^{n+1}, \operatorname{div} v_h) = & \\ \dfrac{1}{\Delta t}\left[(z_h^n(\delta g) \circ \chi_h^{u,n}(g), v_h) - \Delta t \cdot (\nabla u_h^n(\chi_h^{u,n}(g)) \cdot z_h^n(\delta g), v_h)\right], & \forall v_h \in V_h \\ \left(\operatorname{div} z_h^{n+1}, q_h\right) = 0, & \forall\, q \in Q_h \\ z_h^{n+1} = \Pi_h\left(\delta g^{n+1}\right), & \Gamma^s \\ z^{n+1} = 0, & \Gamma_{in} \\ (z_h^0, v_h) = (\Pi_h(\delta u_0), v_h), & \forall v_h\ \in V_h \end{cases} \tag{30}$$

Remark 5 *The linearized system involves the term $\nabla u_h^n(\chi_h^{u,n}(g)$, i.e the gradient of the velocity fied u_h^n evaluated at the characteristic foot $\chi_h^{u,n}(g)$. Depending on the choice of approximation spaces, this term may be more or less well captured.*

Remark 6 *Both non-linear and linearized system can be performed advancing in time :*

- *We first compute the caracteristic feet,*
- *we then evaluate the different r.h.s involved in the generalized Stokes systems,*
- *finally, we solve the linear systems.*

4.1.2 Linearized subcycling backtracking

We recall that the foot of a characteristic curve $\chi(z)$ associated to vector field z is given by, χ_m where

$$\begin{array}{ll} \chi_{i+1} = \chi_i - \Delta t_i \cdot z(\chi_i), & \displaystyle\sum_{i=1}^{m} \Delta t_i = \Delta t \\ \chi_0 = x, & x \in\ \Omega \end{array} \tag{31}$$

Hence, the derivative of the characteristic foot is given by the formula $\delta\chi \stackrel{\text{def}}{=} \dfrac{D\chi(z)}{Dz} \cdot \delta z = \dfrac{D\chi_m(z)}{Dz} \cdot \delta z$, with

$$\begin{cases} \delta\chi_{i+1} = \delta\chi_i - \Delta t_i \cdot \delta z(\chi_i) - \Delta t_i \cdot (\nabla z(\chi_i) \cdot \delta\chi_i), & \\ \delta\chi_1 = -\Delta t_0 \cdot \delta z(x), & x \in\ \Omega \end{cases} \tag{32}$$

Then we get,

$$\left(\frac{D(u_h^n \circ \chi_h^n(g))}{Dg}\right) \cdot \delta g = z_h^n(\delta g) \circ \chi_h^n(g) + \nabla u_h^n(\chi_h^n(g)) \cdot \delta\chi_{m,h}^n \tag{33}$$

where

$$\begin{cases} \delta\chi_{i+1,h}^n = \delta\chi_{i,h}^n - \Delta t_i \cdot z_h^n(\chi_{i,h}^n) - \Delta t_i \nabla u_h^n(\chi_{i,h}^n) \cdot \delta\chi_{i,h}^n, \\ \delta\chi_{1,h}^n = -\Delta t_0 \cdot z_h^n(x), \qquad\qquad x \in \Omega \end{cases} \tag{34}$$

We finally need to solve the following linearized problem,

$$\begin{cases} \dfrac{1}{\Delta t}(z_h^{n+1}, v_h) + \nu\left(\nabla z_h^{n+1}, \nabla v_h\right) - (q^{n+1}, \operatorname{div} v_h) = \\ \dfrac{1}{\Delta t}\left[(z_h^n(\delta g) \circ \chi_h^{u,n}(g), v_h) + (\nabla u_h^n(\chi_h^{u,n}(g)) \cdot \delta\chi_h^n, v_h)\right], \quad \forall v_h \in V_h \\ \left(\operatorname{div} z_h^{n+1}, q_h\right) = 0, \qquad\qquad \forall\, q \in Q_h \\ z_h^{n+1} = \Pi_h\left(\delta g^{n+1}\right), \qquad\qquad \Gamma^s \\ z^{n+1} = 0, \qquad\qquad \Gamma_{in} \\ (z_h^0, v_h) = (\Pi_h(\delta u_0), v_h), \qquad\qquad \forall v_h \ \in V_h \end{cases} \tag{35}$$

Remark 7 *Here we choose to approximate the velocity using Lagrangian $\mathbb{P}_1$ finite elements. Then on each simplex, the gradient ∇u_h is constant, that means that the gradient of u_h is discontinuous at each simplex interfaces. We may get troubles if one of the points $\chi_{i,h}$ is located on faces of a given tetraedron K_h. This may almost never occur, but we need to pay attention on these singularities that may produce numerical artefacts on the solution of the linearized discrete system.*
If we choose a higher-order finite element method, these singularities may disappear.

Remark 8 *In order to avoid large vector field storage, the computation of the linearized characteristic direction $\delta\chi_h^n$ can be performed in the routine which build the non-linear characteristic curves. In this case, the intermediate points $\chi_{i,h}^n$ do not need to be stored.*

4.2 Linearized adaptative backtracking

In the case, of the adaptative algorithm, the foot is still given by, χ_m, with

$$\begin{array}{ll} \chi_{i+1} = \chi_i(z) - \Delta t_i(z) \cdot z(\chi_i), & \\ \chi_0 = x, & x \in \Omega \end{array} \tag{36}$$

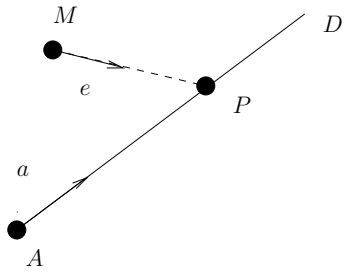

Figure 1: 2D projection along a given vector

The major difference is that m and $\{\Delta t_i\}_{1\leq i\leq m}$ depends on the transport field z, which might add some extra computations and differentiability questions.

4.2.1 2D case

For the sake of simplicity, we first explain how we might perform the derivation of the adaptative characteristic scheme with respect to the vector field $z(.)$ on a 2D case.
Let us consider, a line $D(A,a)$ in $\mathbb{R}^2$ defined by a point $A(x_1^A, x_2^A)$ and a vector $a(a1,a2)$ where the coordinates are given in the orthonormal euclidian frame (O,i,j). Considering a point $M(x_1^M, x_2^M)$ in $\mathbb{R}^2$, we look for the projection point P of point M on D along a vector $e \in \mathbb{R}^2$, which can be identified as $P = P_{\mathbb{R}^2}((A,a),e;M)$ (see Fig. 1).

Lemma 1 *Assuming that vector e is not colinear to a, then the projection point P is determinated by the following expression,*

$$OP = OM + \left(\frac{MA \wedge a}{e \wedge a}\right) \cdot e \tag{37}$$

Lemma 2 *Assuming that vector e is not colinear to a, the derivative of the projection point $P = P_{\mathbb{R}^2}((A,a),e;M)$ considered as a function of the projective direction e is continuous and is given by the following expression,*

$$\frac{d(OP)}{de} = \left(\frac{MA \wedge a}{(e \wedge a)^2}\right) \cdot \left[(e \wedge a)\,\mathrm{I} + e \cdot (a^{\perp})^*\right] \tag{38}$$

with $a^* = (a_1, a_2)$ and $(a^{\perp})^* = (-a_2, a_1)$.

Proof : It can be easily obtained using the cartesian coordinates,

$$\begin{aligned}\frac{d}{de_i}\left(\frac{e_j}{e_1a_2 - e_2a_1}\right) &= \frac{(e_1a_2 - e_2a_1)\delta i,j + (-1)^i e_j a_{i+1}}{(e_1a_2 - e_2a_1)^2} \\ &= \frac{1}{(e \wedge a)^2}\left[(e \wedge a)\,\mathrm{I} + e \cdot (a^{\perp})^*\right]\end{aligned}$$

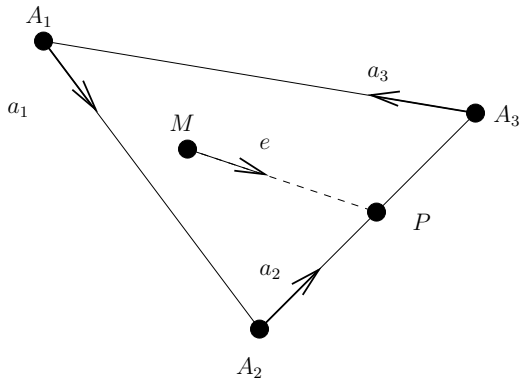

Figure 2: Projection on the edge of a triangle

□

Let us return to the case of the projection of a point on the edge of a triangle along a given direction. We consider a triangle defined by 3 points (A_1, A_2, A_3).

Lemma 3 *Let χ stands for the projection of the point $M \in \mathbb{R}^2$ on the edge of the triangle (A_1, A_2, A_3) along the vector $e \in \mathbb{R}^2$. Then, $\chi(e)$ is differentiable with respect to $e \in \mathbb{R}^2 \setminus \{e_{A_1}, e_{A_2}, e_{A_3}\}$, where e_{A_i} is such that*

$$\chi(e_{A_i}) = A_i$$

Proof : We may only check whether, $\chi(.)$ is differentiable at point e_{A_i} for $i \in [1, 3]$. We consider $e_{A_i^+}$ and $e_{A_i^-}$, such that

- $\chi(e^+_{A_i}) = A_i^+$ and $\chi(e^+_{A_i}) \in]A_i, A_{i+1}[$
- $\chi(e^-_{A_i}) = A_i^-$ and $\chi(e^+_{A_i}) \in]A_{i-1}, A_i[$

Using $\chi(e^+_{A_i}) = \chi(e^-_{A_i})$, we get

$$\left(\frac{MA_i \wedge a_i}{e \wedge a_i}\right) = \left(\frac{MA_{i+1} \wedge a_{i+1}}{e \wedge a_{i+1}}\right) = \rho$$

Thus we get that the jump of derivatives at point e_{A_i} is given by the following expression,

$$\begin{aligned} \frac{d\chi}{de}(e^-_{A_i}) - \frac{d\chi}{de}(e^+_{A_i}) &= \rho \cdot \left[\frac{e_{A_i} \cdot (a_i^\perp)^*}{e_{A_i} \wedge a_i} - \frac{e_{A_i} \cdot (a_{i+1}^\perp)^*}{e_{A_i} \wedge a_{i+1}}\right] \\ &\neq 0, \text{ for } a_i \neq a_{i+1} \end{aligned}$$

This last identity points out the lack of continuity of the derivative $\frac{d\chi}{de}$ for directions pointing towards the vertices of the triangle. □

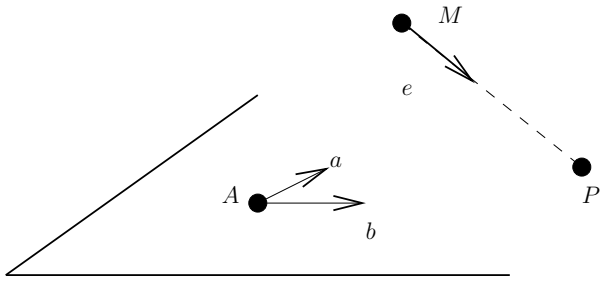

Figure 3: 3D projection

4.2.2 3D case

The previous analysis can be easily extended to the 3D case. We may omit the proof of the following results, since it may be established using the same techniques used in the 2D case.

Lemma 4 *We consider a plan $D(A,a,b)$ defined by a point $A \in \mathbb{R}^3$ and two non-colinear vectors $(a,b) \in \mathbb{R}^3 \times \mathbb{R}^3$ (i.e $a \wedge b \neq 0$). The projection point P of a point $M \in \mathbb{R}^3$ on the plan (A,a,b) along direction given by the vector $e \in R^3$ exists as long as e is not parallel to the plan D (i.e $(e \wedge a) \cdot b \neq 0$) and is given by the following expression,*

$$OP = OM + \left(\frac{(MA \wedge a) \cdot b}{(e \wedge a) \cdot b} \right) \cdot e \tag{39}$$

Lemma 5 *The point P as a function of the vector $e \in \mathbb{R}^3$ is differentiable and its derivative is given by the following identity,*

$$\frac{d(OP)}{de} = \left(\frac{(MA \wedge a) \cdot b}{((e \wedge a) \cdot b)^2} \right) \cdot \left[((e \wedge a) \cdot b) \cdot \mathrm{I} - e \cdot [a \wedge b]^* \right] \tag{40}$$

We consider a tetraedron $K \in \mathbb{R}^3$ defined by the points (A_1, A_2, A_3, A_4). For a point $M \in \mathbb{R}^3$ inside K, the point P stands for the projection of M on ∂K along the direction given by the vector $e \in \mathbb{R}^3$. The following result is an easy consequence of the previous analysis,

Theorem 3 *The point P as a function of $e \in \mathbb{R}^3$ is differentiable on the set $\mathbb{R}^3 \setminus \mathcal{I}$ where the set $\mathcal{I}$ is such that for $e \in \mathcal{I}$, then*

$$P(e) \in \textit{edge of } \partial K$$

4.3 Implementation using barycentric coordinates

In this section, we will describe how the projection step involved in the adaptative backtracking characteristic scheme has been implemented using barycentric coordinates. This will be

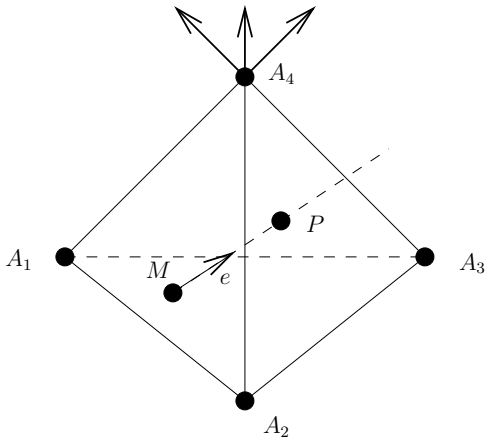

Figure 4: Projection on the edge of a tetrahedron

the basis for deriving an implementable linearized discrete scheme for the Navier-Stokes system.

4.3.1 Projection using barycentric coordinates

Let us go back to the problem of projecting a point inside a tetraedron on its boundary along a given direction. This problem is usually refered in the Computer Graphics litterature as the Ray Tracing problem. It has been proven, that an elegant and efficient way of solving this problem is the use of barycentric coordinates.

Definition 1 *Let us consider a tetraedron $K \in \mathbb{R}^3$ defined by the points (A_1, A_2, A_3, A_4). For a point $M \in \mathbb{R}^3$ inside K, the barycentric coordinates $\{bx_i\}_{1\leq i\leq 4}$ of M are defined as the solution of the following system,*

$$\begin{cases} \sum_{j=1}^{4} bx_j \cdot c(.,j) = x(.) \\ \sum_{j=1}^{4} bx_j = 1 \end{cases} \tag{41}$$

where $\{c(i,j)\}_{1\leq i\leq 3, 1\leq j\leq 4}$ stands for the cartesian coordinates of the vertex A_j of the tetraedron K. $\{x(i)\}_{1\leq i\leq 3}$ stands for the cartesian coordinates of M.

Lemma 6 *If the tetraedron K is not degenerated, then the barycentric coordinates of M are uniquely determinated and $0 \leq bx_j \leq 1,\ 1 \leq j \leq 4$. Furthermore, $\exists i / bx_i = 0 \Longleftrightarrow M \in \partial K$.*

Remark 9 *Using matrix notation, we can solve the following problem*

$$\begin{bmatrix} c(1,1) & c(1,2) & c(1,3) & c(1,4) \\ c(2,1) & c(2,2) & c(2,3) & c(2,4) \\ c(3,1) & c(3,2) & c(3,3) & c(3,4) \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} bx_1 \\ bx_2 \\ bx_3 \\ bx_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Inside NSI3, this system is solved using the substitution,

$$bx_4 = 1 - bx_1 - bx_2 - bx_3$$

setting,

$$C = \begin{bmatrix} c(1,1) - c(1,4) & c(1,2) - c(1,4) & c(1,3) - c(1,4) \\ c(2,1) - c(2,4) & c(2,2) - c(2,4) & c(2,3) - c(2,4) \\ c(3,1) - c(3,4) & c(3,2) - c(3,4) & c(3,3) - c(3,4) \end{bmatrix}$$

and

$$bx = \begin{bmatrix} bx_1 \\ bx_2 \\ bx_3 \end{bmatrix} \quad d = \begin{bmatrix} x_1 - c(1,4) \\ x_2 - c(2,4) \\ x_3 - c(3,4) \end{bmatrix}$$

we may solve,

$$C \cdot bx = d \tag{42}$$

In the sequel, we may also need to express vectors in barycentric form thanks to the following definition,

Definition 2 *Let us consider a tetraedron $K \in \mathbb{R}^3$ defined by the points (A_1, A_2, A_3, A_4). For a vector $a \in \mathbb{R}^3$, the barycentric coordinates $\{ba_i\}_{1 \leq i \leq 4}$ are defined as the solution of the following system,*

$$\begin{cases} \sum_{j=1}^{4} ba_j \cdot c(.,j) = a(.) \\ \sum_{j=1}^{4} ba_j = 0 \end{cases} \tag{43}$$

where $\{c(i,j)\}_{1 \leq i \leq 3, 1 \leq j \leq 4}$ stands for the cartesian coordinates of the vertex A_j of the tetraedron K. $\{a(i)\}_{1 \leq i \leq 3}$ stands for the cartesian coordinates of a.

Remark 10 *Inside NSI3, the barycentric coordinates of a vector are found by substitution,*

$$ba_4 = -ba_1 - ba_2 - ba_3$$

setting,

$$ba = \begin{bmatrix} ba_1 \\ ba_2 \\ ba_3 \end{bmatrix} \quad a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

we may solve,

$$C \cdot ba = a$$

The matrix C enters the definition of the affine transform between reference and actual tetrahedra inside the finite element framework. Hence its inverse C^{-1} is usually stored for all the mesh avoiding by the way, local computation.

Let us come back to the purpose of this section, i.e the computation using barycentric coordinates of the projection of a point M inside a tetrahedron K on its faces along a given direction vector e. This problem can be expressed as follow,

Problem 1 *Given a point M inside a simplex K of barycentric coordinates bx and a vector e with barycentric coordinates be , find $\rho \in \mathbb{R}^+$ such that*

$$\prod_{j=1}^{4} b_j(x - \rho e) = 0 \tag{44}$$

$$b_j(x - \rho \cdot e) \geq 0,\ 1 \leq j \leq 4 \tag{45}$$

Lemma 7 *With the nondegeneracy assumptions on K, the following identity holds true*

$$b(x - \rho e) = bx - \rho \cdot be \tag{46}$$

Lemma 8 *There exists a unique step $\rho^* \in \mathbb{R}^+$ and at least one index $m \in [1, 4]$ such that*

$$\rho^* = \left(\frac{(bx)_m}{(be)_m} \right),$$
$$b_j(x - \rho^* \cdot e) \geq 0, \quad 1 \leq j \leq 4$$

Remark 11

Inside the code NSI3, this step is performed by trial and error, by changing the index m and checking the above conditions until it works, i.e

1. *Set $m = 1$ and choose a tolerance parameter $\varepsilon > 0$,*
2. *Compute $\rho^m = \left(\frac{(bx)_m}{(be)_m} \right)$,*
3. *if $b_j(x - \rho^m \cdot e) < -\varepsilon$ for $j \neq m$, go to step 5.,*
4. *else set $\chi = x - \rho^m \cdot e$ and STOP.*
5. *Set $m \leftarrow m + 1$ and go back to step 2.*

4.3.2 Derivative of the projection using barycentric coordinates

Let us go back to the original problem of computing the derivative with respect to the convective velocity field of the characteristic foot associated to the adaptative backtracking scheme. For this purpose, we need to differentiate the previous projection formula with respect to the vector $e \in \mathbb{R}^3$.

Lemma 9 *Let χ stands for the cartesian coordinates of the projection of point M on ∂K. The coordinate χ is differentiable for $e \in \mathbb{R}^3 \setminus \mathcal{I}$ and,*

$$\left(\frac{d(b\chi)}{de}\right) \cdot \delta e = -\rho^*(e) \cdot b(\delta e) + \left(\rho^*(e) \cdot \frac{(b(\delta e))_m}{(be)_m}\right) \cdot be \tag{47}$$

where $\mathcal{I}$ is such that for $e \in \mathcal{I}$

$$\chi(e) \in \textit{edge of } \partial K$$

Proof : Using the identity $b\chi = bx - \rho^* \cdot be$, we get

$$\left(\frac{d(b\chi)}{de}\right) \cdot \delta e = -\left(\frac{d\rho^*(e)}{de} \cdot b(\delta e)\right) \cdot be - \rho^* \cdot b(\delta e) \tag{48}$$

Using $\rho^* = \left(\frac{(bx)_m}{(be)_m}\right)$, we deduce that

$$\begin{aligned} \left(\frac{d\rho^*(e)}{de} \cdot b\right) \cdot \delta e &= -\frac{bx_m \cdot (b(\delta e))_m}{(be)_m^2} \\ &= -\rho^*(e) \cdot \left(\frac{(b(\delta e))_m}{(be)_m}\right) \end{aligned}$$

from which we deduce the derivative identity. □ A straightforward consequence is the following result,

Lemma 10

$$\frac{d\chi}{de} \cdot \delta e = -\rho^*(e) \cdot \left[\delta e - \left(\frac{(b(\delta e))_m}{(be)_m}\right) \cdot e\right] \tag{49}$$

Then, once the projection on the edge of K has been done, we can compute the linearized characteristic foot in the direction δe, for the knowledge of the barycentric coordinates of the initial point, the projection step $\rho^*(e)$ and the barycentric coordinates of the perturbation direction δe. In fact, only the barycentric coordinates of index m play a role, where m is the index solution of the algorithm described in remark 11.

4.3.3 linearization of the discrete scheme inside NSI3

The previous steps allows us to linearize the full discrete approximation of the Navier-Stokes system used inside the computational research code NSI3. This scheme has the special feature of using an adaptative backtracking characteristic algorithm.

- **Navier-Stokes full discrete scheme** :
  we look for $(u_h^{n+1}, p_h^{n+1}) \in X_h^3 \times Q^h$ solution of the following system,

$$\begin{cases} \frac{1}{\Delta t}(u_h^{n+1}, v_h) + \nu(\nabla u_h^{n+1}, \nabla v_h) - (p_h^{n+1}, \operatorname{div} v_h) = \frac{1}{\Delta t}(u_h^n \circ \chi_h^{u,n}, v_h), & \forall\, v_h \in V_h \\ (\operatorname{div} u_h^{n+1}, q_h) = 0, & \forall\, q_h \in Q_h \\ u_h^{n+1} = \Pi_h g^{n+1}, & \Gamma^s \\ u_h^{n+1} = \Pi_h u_\infty^{n+1}, & \Gamma_{in} \end{cases} \tag{50}$$

  where $\chi_h^{u,n} = \chi_{m,h}^{u,n}$ stands for an approximation of the characteristic foot coming from point $x \in \Omega$ computed using the following scheme,

- **Adaptative backtracking scheme** :

$$\begin{aligned} &\chi_{i+1,h}^{u,n} = \chi_{i,h}^{u,n} - \rho_i \cdot u_h^n(\chi_{i,h}^{u,n}), \\ &\chi_{0,h}^{u,n} = x, \qquad\qquad x \in K_h \end{aligned} \tag{51}$$

  with

$$\rho_i = \left(\frac{(b\chi_{i,h}^{u,n})_{m_i}}{(bu_h^n(\chi_{i,h}^{u,n}))_{m_i}} \right), \quad \text{in the frame } K_h^i \tag{52}$$

Then, it is possible to establish the structure of the (LDP) problem,

- **Linearized discrete problem** We look for (z_h^n, q_h^n) solution of the following linear system,

$$\left\{\begin{array}{ll}
\dfrac{1}{\Delta t}(z_h^{n+1}, v_h) + \nu\left(\nabla z_h^{n+1}, \nabla v_h\right) - (q^{n+1}, \operatorname{div} v_h) = & \\
\dfrac{1}{\Delta t}\left[(z_h^n(\delta g) \circ \chi_h^{u,n}(g), v_h) + \left(\nabla u_h^n(\chi_h^n(g)) \cdot \left(\dfrac{D\chi_h^{u,n}(z)}{Dz} \cdot z_h^n(\delta g)\right), v_h\right)\right], & \\
\forall v_h \in V_h & \\
\left(\operatorname{div} z_h^{n+1}, q_h\right) = 0, & \\
\forall\, q \in Q_h & \\
z_h^{n+1} = \Pi_h\left(\delta g^{n+1}\right), & \Gamma^s \\
z^{n+1} = 0, & \Gamma_{in} \\
(z_h^0, v_h) = (\Pi_h(\delta u_0), v_h), & \\
\forall v_h \in V_h &
\end{array}\right. \tag{53}$$

with the following characteristic derivative scheme,

- **Linearized adaptative backtracking scheme**

$$\frac{D\chi_h^{u,n}}{Du_h^n}(\chi_h^{u,n}) \cdot z_h^n = (\delta\chi_{m,h}^{u,n})) \cdot z_h^n \tag{54}$$

with

$$\left\{\begin{array}{rcl}
(\delta\chi_{i+1,h}^{u,n}) \cdot z_h^n & = & (\delta\chi_{i,h}^{u,n}) \cdot z_h^n - [(\delta\rho_i) \cdot z_h^n] \cdot u_h^n(\chi_{i,h}^{u,n}) \\
 & & -\rho_i \cdot \left[\nabla u_h^n \circ \chi_{i,h}^{u,n}\right] \cdot (\delta\chi_{i,h}^{u,n}) \cdot z_h^n - \rho_i \cdot z_h^n(\chi_{i,h}^{u,n}), \\
\delta\chi_{1,h}^{u,n} & = & -[(\delta\rho_0) \cdot z_h^n] \cdot u_h^n(x) - \rho_0 \cdot z_h^n(x), \quad x \in K_h
\end{array}\right. \tag{55}$$

with

$$\rho_i = \left(\frac{(b\chi_{i,h}^{u,n})_{m_i}}{(bu_h^n(\chi_{i,h}^{u,n}))_{m_i}}\right), \text{ in the frame } K_h^i \tag{56}$$

and

$$(\delta\rho_i) \cdot z_h^n = -\rho_i \left(\frac{(bz_h^n(\chi_{i,h}^{u,n}))_{m_i}}{(bu_h^n(\chi_{i,h}^{u,n}))_{m_i}}\right), \text{ in the frame } K_h^i \tag{57}$$

Remark 12 *This algorithm is only valid if we assume that the projected points on each simplex do not cross the edge of ∂K. This is a strong hypothesis, but the probability for this to occur is weak.*

Remark 13 *A much stronger obstruction for the last algorithm to perform well inside NSI3 is that we only have a $\mathbb{P}^1$ approximation for the velocity, this means that the gradient ∇u_h is discontinuous on each side of the simplex $K \in \mathcal{T}_h$. This causes the adaptative backtracking scheme to be non-differentiable inside NSI3. If we choose a higher-order finite element approximation, this will not be the case, even if problems at the edge of the vertex can still remain. We will show in the sequel the kind of numerical problems that arises with such an implementation inside NSI3.*

4.4 Discrete linearized system

As described previously, we may be interested in a different procedure which consists in,

1. linearizing the state problem,
2. then discretizing the linearized problem.

This leads to the discrete linearized problem (DLP). Let us describe this procedure for the Navier-Stokes problem.
We have already establish the structure of the continous linearized problem with respect to Dirichlet boundary conditions. We set,

$$(z(\delta g), q(\delta g)) = \left(\frac{D(u,p)}{Dg}\right) \cdot (\delta g)$$

stand for the state derivatives with respect to the parameter g in the perturbation direction δg. The directional derivatives are solution of the following linear evolution system,

$$\begin{cases} \partial_t z + \nabla u \cdot z + \nabla z \cdot u - \nu \Delta z + \nabla q = 0, & \Omega \\ \operatorname{div} z = 0, & \Omega \\ z = 0, & \Gamma^s \times (0,T) \\ \sigma(z,q) \cdot n = 0, & \Gamma_{out} \\ z = \delta g, & \Gamma_{in} \\ z(t=0) = 0, & \Omega \end{cases} \tag{58}$$

The main issue is the choice of the time discretization for the above system.
Let us introduce as previously the solution χ^u of the following Cauchy problem,

$$\begin{cases} \frac{d}{d\tau}\chi^u(\tau;t,x) = u(\tau, \chi^u(\tau;t,x)), \quad \tau \in (0,t) \\ \chi^u(\tau = t;t,x) = x \end{cases} \tag{59}$$

Using the chain rule, it can be easily stated that the following identity holds true,

$$\frac{d}{dt}\left(z \circ \chi^u\right) = \partial_t z + \nabla z \cdot u \tag{60}$$

Hence the linearized system can be viewed as a combinaison of,

1. a Stokes operator,

$$\begin{cases} -\nu\Delta z + \nabla q \\ \operatorname{div} z = 0 \end{cases}$$

2. a linear advection operator,

$$\frac{d}{dt}(z \circ \chi^u)$$

3. and a reaction operator,

$$\nabla u \cdot z$$

The last operator is the key problem in the derivation of efficient approximation scheme for the linearized Navier-Stokes system, since depending on the sign and the size of ∇u, this term may add a non-positive dissipation contribution resulting in an unstable numerical scheme.
For the time being, as reported in [10], there is no efficient strategies in order to stabilize the reaction term $\nabla u \cdot z$. For that reason, we suggest to use an explicit development of this term in the discretization. Then using a first-order characteristic scheme coupled with the same finite element strategy used for the state problem, we obtain the following full discrete problem,

$$\begin{cases} \frac{1}{\Delta t}(z_h^{n+1}, v_h) + \nu(\nabla z_h^{n+1}, \nabla v_h) - (q_h^{n+1}, \operatorname{div} v_h) = \frac{1}{\Delta t}(z_h^n \circ \chi_h^{u,n}, v_h) & \\ -(\nabla u_h^n \cdot z_h^n, v_h), & \forall\, v_h \in V_h \\ (\operatorname{div} z_h^{n+1}, q_h) = 0, & \forall\, q_h \in Q_h \\ z_h^{n+1} = \Pi_h(\delta g^{n+1}), & \Gamma^s \\ z_h^{n+1} = 0, & \Gamma_{in} \\ (z_h^0, v_h) = 0, & \forall\, v_h \in V_h \end{cases} \tag{61}$$

Remark 14

1. *Such a scheme may be used, as long as the mesh is enough refined in the region where the gradient of u_h^n is large. These regions may correspond to the existence of boundary layers near the walls.*

2. *In the last system, we do not need to compute the caracteristic feet $\chi_h^{u,n}$, since it may be furnished by early state computation steps.*

3. *The above scheme has the advantage to be easily implementable. Nevertheless, the cost functional gradient approximation computed with such a linearized state may not be consistent with the cost functional approximation. This causes the optimization procedure to possibily fail.*

5 Implementation

In this section, we describe some details about the implementation of the whole optimization structure, including the Navier-Stokes solvers, the optimization routines and the parallel direct mode strategy for the gradient computation.

5.1 Nonlinear and linearized Navier-Stokes solvers

5.1.1 NSI3

The NSI3 solver is a 3D sequential Incompressible Navier-Stokes equations solver used at INRIA. It is a Fortran 77 code. Its main features are listed below :

- mixed finite element P1-bubble/P1 for the spatial discretisation.
- Semi-lagrangian (Lagrange-Galerkin method) implicit time discretisation
- ALE capable for fluid/structure interaction.

This code is based upon solving at each time step a generalized Stokes problem with a preconditionned Conjugate-Gradient algorithm for the Schur complement system. Setting A_h to be the matrix associated to the operator $\frac{D}{Dt} - \nu\Delta$, and B_h the matrix associated to the operator ∇, the system to be solved may be written as a linear system of the following form :

$$\left\{ \begin{array}{rcl} A_h U_h + B_h P_h & = & F_h \\ B_h^T U_h & = & 0 \end{array} \right.$$

where U_h and P_h are the discrete velocity and pression unknowns. The Uzawa method is based upon writing this system only with the pressure unknown P_h. If we set :

$$U_h = A_h^{-1}[F_h - B_h P_h]$$

the modified system may be written in the following form :

$$M_h P_p = B_h^T A_h^{-1} F_h$$

The matrix $M_h = B_h^T A_h^{-1} B_h$ is called the Schur complement and is symmetric positive definite and well conditionned and thus may be used as a preconditioner. The linear system is then solved using a SSOR-preconditioned CG method. For more informations, see [37] and [39].

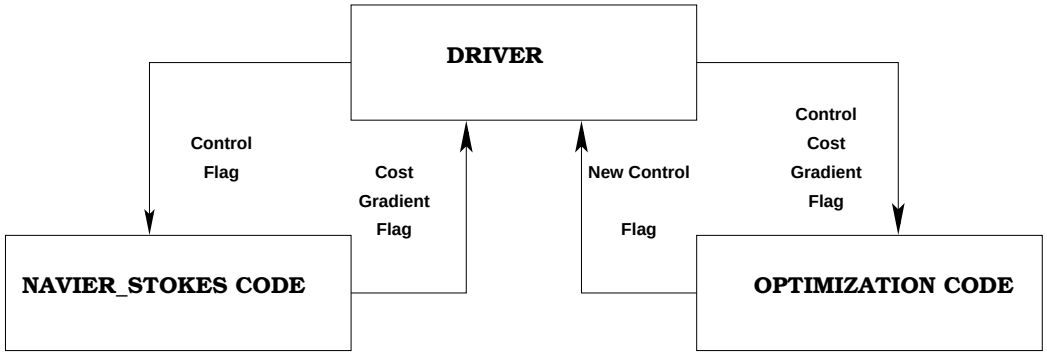

Figure 5: Optimization tool structure

5.1.2 LGNSI2FS

LGNSI2FS is a 2D Lagrange-Galerkin code we developed during our PhD. It includes the main features of NSI3 such as P1-Bubble/P1 mixted formulation, Lagrange-Galerkin method with adaptative backtracking methods and also fluid-structure algorithms. However the subcycling backtracking method is also implemented and the linear system is solved using a standard GMRES method based upon the Sparsekit library developed by Y. Saad[1] with an ILUT preconditioner. The ILUT preconditionner is an Incomplete LU factorisation with a dropping strategy. This means that, when assembling the ILU matrix used as a preconditioner, elements under a given drop tolerance are ignored. The complete description of this preconditioning algorithm may be found in [41].

5.2 Optimization routines

As described in section 2, we use gradient based methods in order to solve the minimization problem (2). There exists a huge amount of research studies concerning the development of efficient optimization algorithms and routines [36, 5]. The major effort is to build optimization tools that can be used by any one without deep knownledge of the underlying optimization algorithms. These tools are very pratical, since the parameter of the routines have been tuned to reach optimal efficiency, e.g globalization issue using line search is one of the hardest part to tune.
For these reasons, we chose to work with two optimization routines that have been benchmarked by the optimization community for academic and pratical applications. We have used the following routines :

[1] http://www.cs.umn.edu/research/arpa/SPARSKIT/sparskit.html

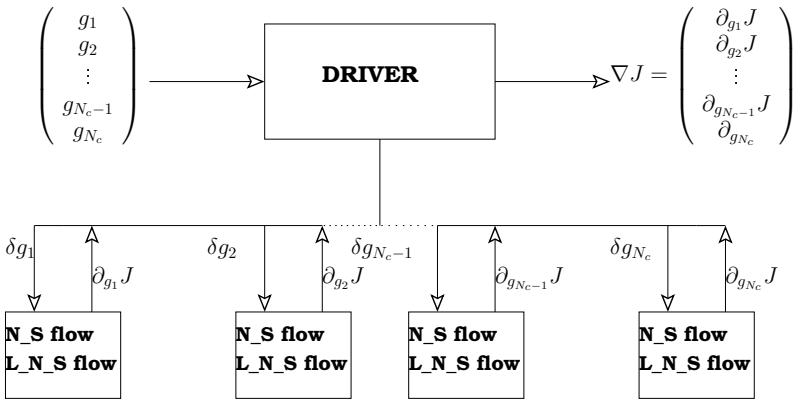

Figure 6: Parallel strategy for the gradient computation

- The CGPLUS [2] routine based on the conjugate gradient method described in [21] with a very efficient line search procedure.
- The L-BFGS-B[3] routine which is based on a bound constraint quasi-Newton method with BFGS update rule described in [7, 43].

As described in Fig. 5, we only need to furnish the current control, the value of the cost function and its associated gradient for the optimization routines to perform. Hence we can easily change either the fluid solvers or the optimization solvers. As a consequence, we have at our disposal an evolutive tool that can be easily updated in the future.

5.3 Parallel direct mode strategy

As stated before, we chose to compute the cost-function gradient using the linearized state computation using the linearized discrete Navier-Stokes system described in section 4. This means that the number of linearized systems to solve is equal to the cardinal of the control space. This drawback is balanced by the fact that the linearized system can be advanced forward in time with the nonlinear state computation (see Remark 8).
We took advantage of this situation by adopting a parallel computation of the linearized state (z_h, q_h). Its means that we need to spawn N_c linearized state computations corresponding to the N_c directions of the control state space. For each spawned task, we compute sequentially the non-linear state and the linearized state for every time step (see Fig. 6). This avoids local storage of the characteristic feet and local vertices involved in the characteristic backtracking procedure. This strategy is restricted by the number of processor at our disposal and is considered as a first step towards efficient strategy for the control of fluid systems. However,

[2] http://www-neos.mcs.anl.gov/neos/solvers/UCO:CGPLUS/
[3] http://www-neos.mcs.anl.gov/neos/solvers/BCO:L-BFGS-B/

assuming we have N_c processors, the complexity of this approach is almost equal to the complexity of a non-linear and linearized flow joint computation.

6 Discrete gradient computation

In this section, we shall analyse the accuracy and consistency of the discrete gradient in some simple fluid flow configurations, namely the Poiseuille and the driven cavity flow. For both cases, we compare our discrete gradient to the differential quotient approximation of the gradient.

6.1 Case of Poiseuille flow

Let us first consider an open rectangular fluid domain, where we impose parabolic inflow boundary conditions, i.e (u, p) is solution of the following Navier-Stokes system,

$$\begin{cases} \partial_t u + \nabla u \cdot u - \nu \Delta u + \nabla p = 0, & \Omega \\ \mathrm{div}(u) = 0, & \Omega \\ u(\rho, x, t) = 4\rho(t)x(h-x)/h^2, & \Gamma_{in} \\ \sigma(u,p) \cdot n = 0, & \Gamma_{out} \\ u(t=0) = 4\rho(0)x(1-x)/h^2, & \Omega \end{cases} \tag{62}$$

First let us assume, that $\rho \in \mathbb{R}$ is constant. The solution of the above system is given by:

$$u(x,t) = 4\rho x(h-x)/h^2$$

We shall need in the sequel to compute the derivative of the state variable $(u(\rho), p(\rho)$ with respect to ρ, it is given by the couple (z, q) solution of the following linearized Navier-Stokes system,

$$\begin{cases} \partial_t z + \nabla u \cdot z + \nabla z \cdot u - \nu \Delta z + \nabla q = 0, & \Omega \times (0,T) \\ \mathrm{div}(z) = 0, & \Omega \times (0,T) \\ \sigma(z,q) \cdot n = 0, & \Gamma_{out} \times (0,T) \\ z = 4\delta\rho x(h-x)/h^2, & \Gamma_{in} \times (0,T) \\ z(t=0) = 4\delta\rho x(h-x)/h^2, & \Omega \end{cases} \tag{63}$$

Like the non-linear equation (62), an exact solution of (63) is available :

$$z(x,t) = 4\delta\rho x(h-x)/h^2$$

6.1.1 Flow tracking control problem

We would like to minimize the following optimization problem,

$$J(\rho) = \frac{1}{2} \int_0^T \int_{\Gamma_{out}} |u(\rho) - u_d|^2 \quad d\Gamma dt \tag{64}$$

where u_d is a known velocity field. We can choose, for instance, $u_d = 4\rho_d x(h-x)/h^2$ for a simple tracking optimization. Then, because of the extremly simple shape of the solution, it is possible to compute exactly $J(\rho)$:

$$\begin{aligned} J(\rho) &= \frac{8T}{h^4}(\rho-\rho_d)^2 \int_0^h (x(h-x))^2 \quad dx \\ &= \frac{4T}{15}h(\rho-\rho_d)^2 \end{aligned}$$

In order to apply the conjugate gradient algorithm described in the previous section, we need to compute the derivative of $J(\rho)$ with respect to ρ,

$$(\nabla J(\rho), \delta\rho) = \int_0^T \int_{\Gamma_{out}} (u(\rho)-u_d) z(\delta\rho) \quad d\Gamma dt$$

where $(z(\delta\rho), q(\delta\rho)$ are solutions of the linearized Navier-Stokes system (63), whose solution is given by :

$$z(x,t) = 4\delta\rho x(h-x)/h^2$$

and we have,

$$(\nabla J(\rho), \delta\rho) = \frac{8T}{15}h(\rho-\rho_d)\cdot\delta\rho$$

We may compare, such an expression with the one computed using the linearized flow (z,q) and the one obtained by using finite difference, i.e

$$DJ(\rho,\delta\rho) = \frac{J(\rho+\delta\rho)-J(\rho)}{\delta\rho}$$

Since J is twice differentiable with respect to ρ and $\nabla^2 J(\rho) = \frac{8T}{15}h$, we may expect that,

$$DJ(\rho,\delta\rho) - \nabla J(\rho) = \frac{4T}{15}h\delta\rho \tag{65}$$

Using this equation, we can now check accuracy and consistency of the discrete cost function gradient.

6.1.2 Derivative checking : the intersection algorithm case

Test setup We first use the NSI3 code to compute the Poiseuille flow and its associated linearized flow. The 2D computational domain is represented in Fig. 7. The final computation domain is an elevation of this 2D domain with an elevation parameter of 0.1 in depth. The discretisation was performed using $\Delta x = 1/40$, $\Delta y = 1/40$ and $\Delta z = 0.1$. The overall scheme was integrated over a period of 10 seconds.

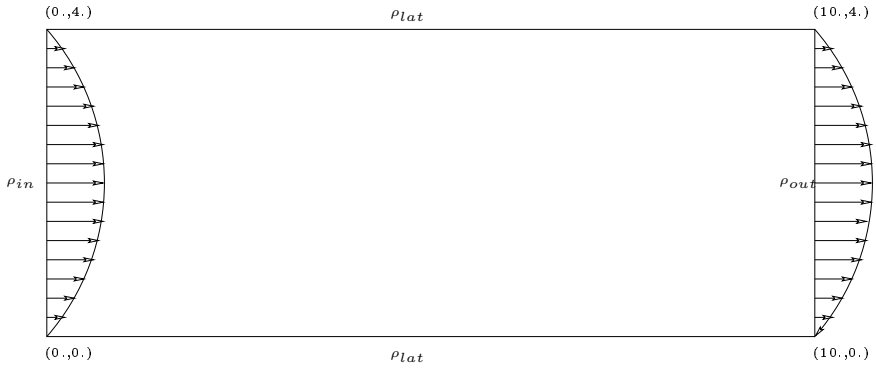

Figure 7: Poiseuille Flow domain

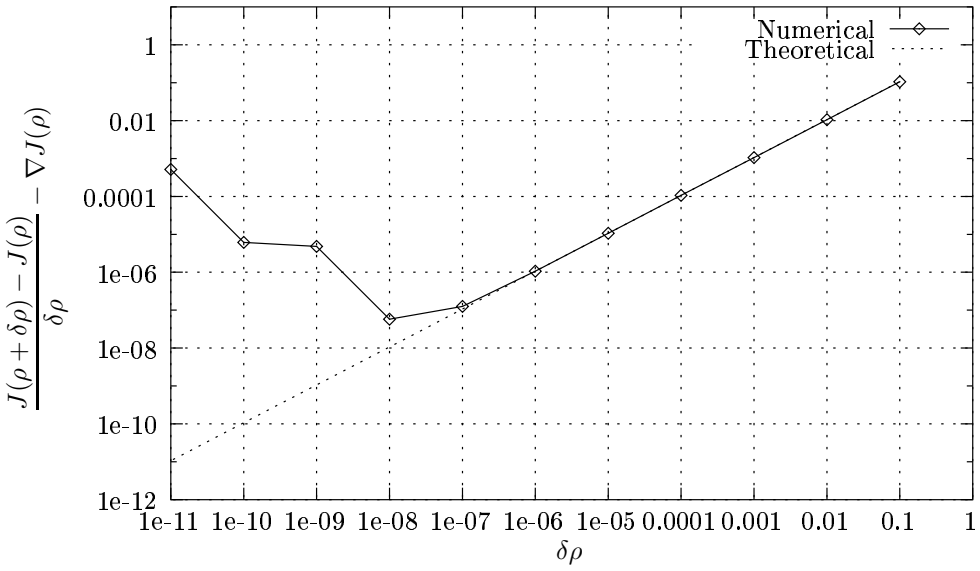

Figure 8: First order convergence of the finite difference gradient for the Poiseuille Flow

First Order LG NSI3 code		
Time Step	Cost	Lin. Cost
0.10	10.6595	21.1319
0.05	10.6595	21.1319

Table 1: Cost and cost function for the first Lagrange-Galerkin method using the NSI3 code

Test results We set here $\rho_d = 0$, $\rho = 1$, and we compare the discrete gradient obtained by the linearized state computation with the finite difference gradient for different perturbation steps. Results are given by Fig. 8. Up to $\delta\rho = 10^{-6}$, the linearized gradient is consistent with the finite difference gradient. It shows that we are indeed solving the exact linearization of the discrete Navier-Stokes system, which guarantees a consistent approximation of the cost function gradient. Furthermore, we recover the linear decrease of the consistency error $DJ(\rho, \delta\rho)$ with respect to $\delta\rho$ up to $\delta\rho = 10^{-6}$ with the correct slope $\frac{4T}{15}h$.
Below the perturbation level $\delta\rho = 10^{-6}$, finite precision computations produce larger gradient finite difference approximations. These quantities are no more relevant for the approximation of the cost function gradient. Table 1 represents the cost function and its gradient for two time steps using the NSI3 code with first order LG using adaptative backtracking :

- The time steps seems irrelevant as far as the costs functions are concerned. This is not surprising since the Poiseuille flow is a stationary flow, but this proves that the characteristics path as well as the linearized state are well computed. In fact, the cost function and its gradient are constant up to 1e-13, which is almost the machine precision in double precision.
- The second remark is that the cost function and its gradient are very well approximated by our code. More precisely, we have :

$$\frac{\nabla J_h^{dt} - \nabla J_{theo}}{\nabla J_{theo}} = 6.0 \times 10^{-5}$$

  This means that we may use the linearized gradient as an approximation of the cost function gradient inside an optimization procedure.

6.1.3 Simple control test case

Now, we try to track a given parabolic profile at the outflow boundary, i.e with $u_d = 16\rho_d x(h-x)/h^2$ with $\rho_d = 5$. This can be done by minimizing the tracking functional introduced previously. We use the conjugate gradient method with the starting point $\rho^0 = 2$ and an input flow $u = 16\rho x(h-x)/h^2$. Convergence was reached after 10 iterations as shown in Fig. 9 and Fig. 10. We found the optimization process very robust with respect to the starting point value. This means that the functional is strongly convex and has a unique optimimum.

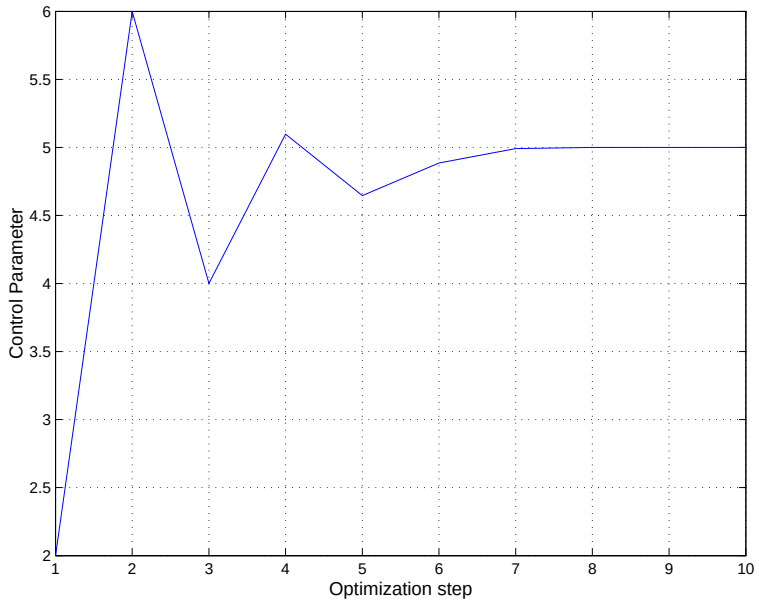

Figure 9: Optimization process for the Poiseuille flow.

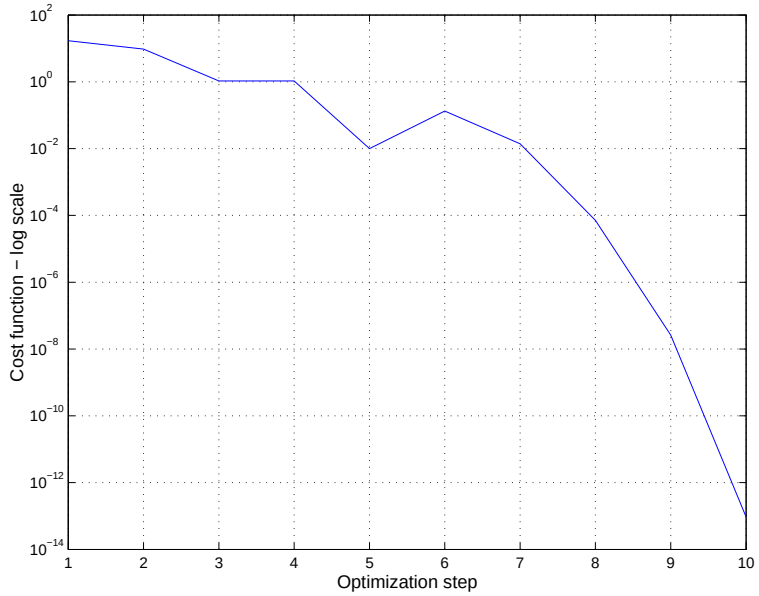

Figure 10: Tracking cost function decrease for the Poiseuille flow.

subcycling backtracking algorithm		
Order	Cost	Lin. Cost
1	10.6522950	21.3045918
2	10.6522943	21.3045887

Exact cost functions	
Cost	Lin. Cost
10.6666666	21.3333333

Table 2: Cost function and its gradient for the first and second order Lagrange-Galerkin methods

6.1.4 derivative checking : case of the subcycling backtracking procedure

We shall perform similar tests using our 2D test code with first and second order characteritics coupled with the subcycling backtracking algorithm. Table 2 and fig. 11 gives the result given by the 2D code. As for the NSI3 code, the finite difference gradient behaves as expected, matching the theoretical slope accurately, but this time numerical errors appear for $\delta\rho < 10^{-5}$.

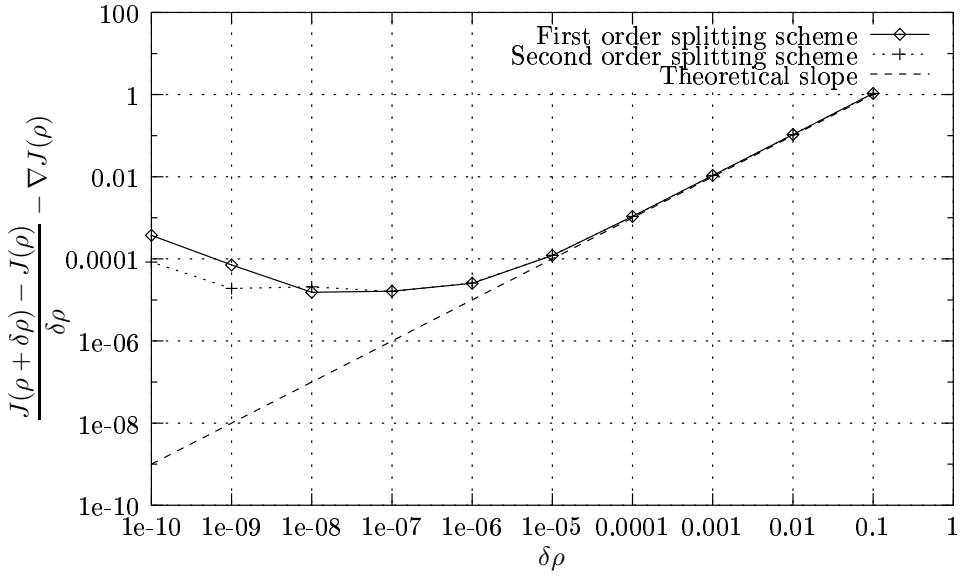

Figure 11: First and second order subcycling backtrackings scheme gradient checking with the theoretical slope for the Poiseuille flow

6.2 The driven cavity

6.2.1 Definition of the problem

Let $\Omega = [0,1] \times [0,1] \times [0,0.1]$ be the computational domain and Ω_h an associated triangulation. We are interested in solving the following problem,

$$\begin{cases} \partial_t u + \nabla u \cdot u - \nu \Delta u + \nabla p = 0, & \Omega \times (0,T) \\ \mathrm{div}(u) = 0, & \Omega \times (0,T) \\ u(.,t) = \rho(t) x(1-x), & \Gamma_1 \times (0,T) \\ u(.,t) = 0. & \Gamma_2 \times (0,T) \end{cases} \tag{66}$$

with $\Gamma_1 = (x, y = 1, z)$, $\Gamma_2 = \partial\Omega - \Gamma_1$. In order to accelerated convergence, $(u_0, p_0) = (u(t = 0), p(t = 0))$ are initialized with the solution of the corresponding Stokes flow :

$$\begin{cases} -\nu \Delta u_0 + \nabla p_0 = 0, & \Omega \times (0,T) \\ \mathrm{div}(u_0) = 0, & \Omega \times (0,T) \\ u(.,t=0) = \rho x(1-x), & \Gamma_1 \times (0,T) \\ u(.,t=0) = 0, & \Gamma_1 \times (0,T) \end{cases} \tag{67}$$

This time, the cost function will be defined as follow :

$$J(\rho) = \frac{1}{2} \int_0^T \int_\Omega |u(\rho)|^2 d\Omega dt \tag{68}$$

which is the integral of the kinetic energy over $[0,T]$. The derivative of $J(\rho)$ follows :

$$(\nabla J(\rho), \delta\rho) = \int_0^T \int_\Omega u(\rho) z(\delta\rho) d\Omega dt \tag{69}$$

6.2.2 Non-differentiability illustration for the adaptative backtracking scheme

In this paragraph, we perform the gradient consistency test for the cavity flow with the code NSI3. As a matter of fact, this test shed some light on the non-differentiability of the characteristic method while using an adaptative backtracking scheme coupled with low order finite elements. Indeed as shown in Fig. 12, for the time steps $dt \in [0.01, 0.1]$, the quantity

$$D_g \stackrel{\text{def}}{=} \frac{(\nabla J_h^{dt} - \mathrm{D}\, J_h^{dt})}{\mathrm{D}\, J_h^{dt}}$$

does not converge to 0 as the finite difference step $\delta\rho$ goes to 0. Actually, it has a non zero asymptotic limit that decrease while decreasing the time step. For large time steps, the characteristic paths intersect increasing numbers of tetrahedra edges. Since we need to evaluate the discontinuous gradient of the reference flow u_h^n at each of the intersection points, this creates numerical errors that grow with time and with the number of characteristic

paths. Decreasing the time step has the effect of reducing the number of intersection with tetrahedra edges, resulting in lower asymptotic limit of the gradient residual D_g. However for a time step $dt = 0.01$, this asymptotic limit is of order 10^{-4} which is still unsatisfactory to validate the discrete gradient. Hence, we have shown that the NSI3 linearized code has poor numerical properties and it was predicted by the theory. In the sequel, we shall use our 2D research code and avoid the use of the adaptative backtracking scheme.

6.2.3 Discrete gradient consistency using the subcycling backtracking algorithm

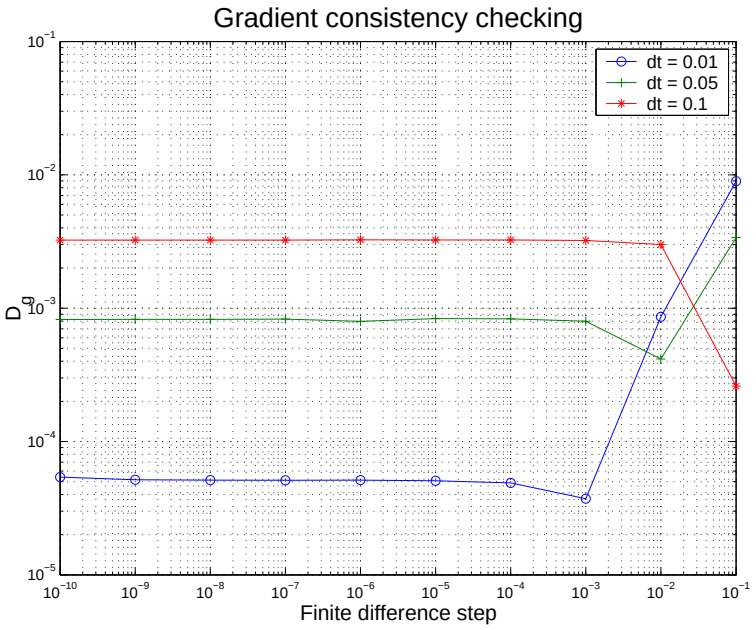

Figure 12: First order Lagrange-Galerkin scheme applied to the driven cavity flow - Gradient consistency checking with NSI3

Numerical settings Let now Ω_h be a regular triangulation of $\Omega = [0,1] \times [0,1]$ with $\Delta x = \Delta y = 1/40$. The right hand side was computed using a 7 points Gauss quadrature. We choose to compute the solution of (66) over a period of $20s$, thus reaching steady state. The characteristics path is computed using the subcycling backtracking algorithm, with a splitting parameter $l_{\text{split}} = 10$.

Remark 15 *The splitting parameter l_{split} was chosen a priori. Problems regarding the choice of this parameter will be discussed later.*

Derivative consistency The procedure for checking the derivative consistency is basically the same as the one used for the Poiseuille flow. Here, we chose to compute the linearized gradient with $\Delta t = 0.05$ over 400 iterations. Fig. 13 gives the first order convergence of the

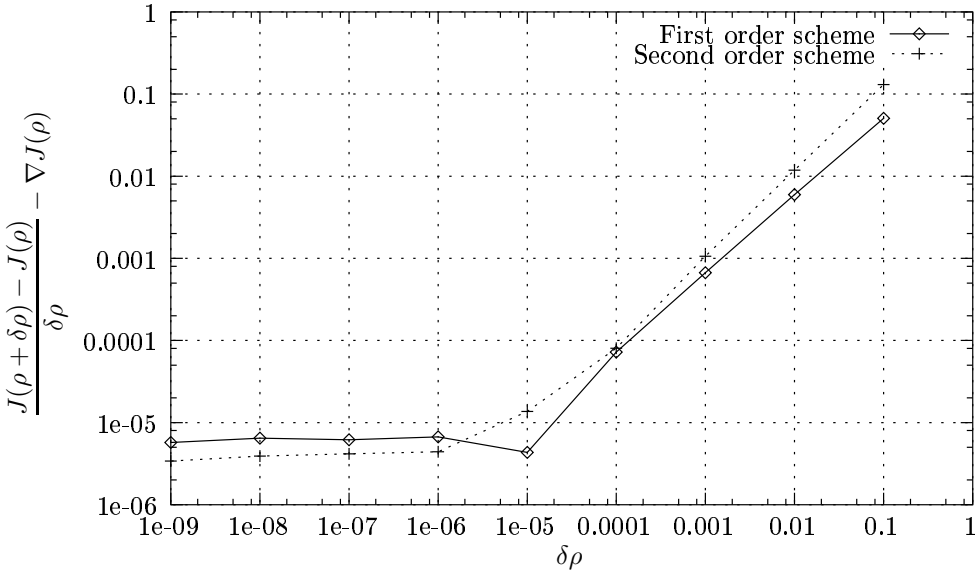

Figure 13: First order and second order Lagrange-Galerkin scheme applied to the driven cavity flow - Gradient checking

finite element gradient for both first order and second order scheme. We can see that both first and second order schemes globally perform the same way. However, for $\delta\rho \geq 10^{-5}$, the first order finite difference derivative gives a better approximation of the derivative than its second order counterpart. Then, for $\delta\rho \leq 10^{-5}$, both schemes give constant result with a better approximation for the second order scheme. Computing the relative errors between the finite difference gradient and the linearized state gradient we get

$$\frac{\nabla J_h^{dt} - \mathrm{D}\, J_h^{dt}}{\mathrm{D}\, J_h^{dt}} = 1.16 \times 10^{-5}$$

for the first order and

$$\frac{\nabla J_h^{dt} - \mathrm{D}\, J_h^{dt}}{\mathrm{D}\, J_h^{dt}} = 2.1 \times 10^{-6}$$

for the second order, from which we may conclude that the discrete gradient is fairly well computed. Table 4 gives the cost function value and its gradient value for both order with $\rho = 1$, and the overall and characteristics path time computation. The second order scheme

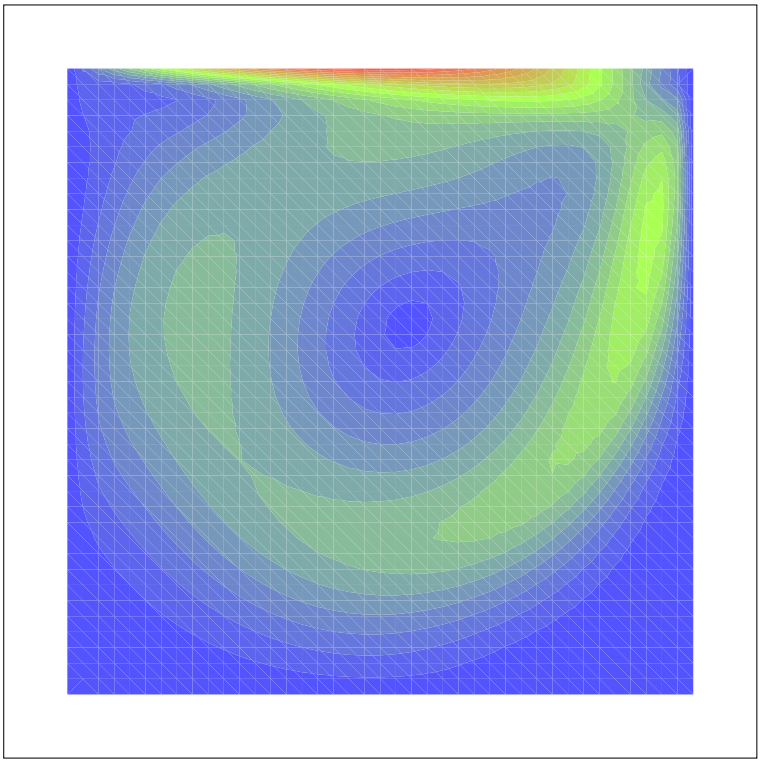

Figure 14: l_2 norm of the driven cavity at steady state

Cost	Cost Grad.	Tot. Calc. Time	Char. Calc. Time
1.018	2.155	30421	7701

Table 3: Reference solution : first order LG scheme $\Delta t = 0.001$

Order	Cost	Cost grad.	Tot. Calc. Time	Char. Calc. Time
1	0.859	1.604	959	351
2	1.013	2.168	1508	765

Table 4: First and second order LG schemes comparison with $rho = 1$ and $\Delta t = 0.05$

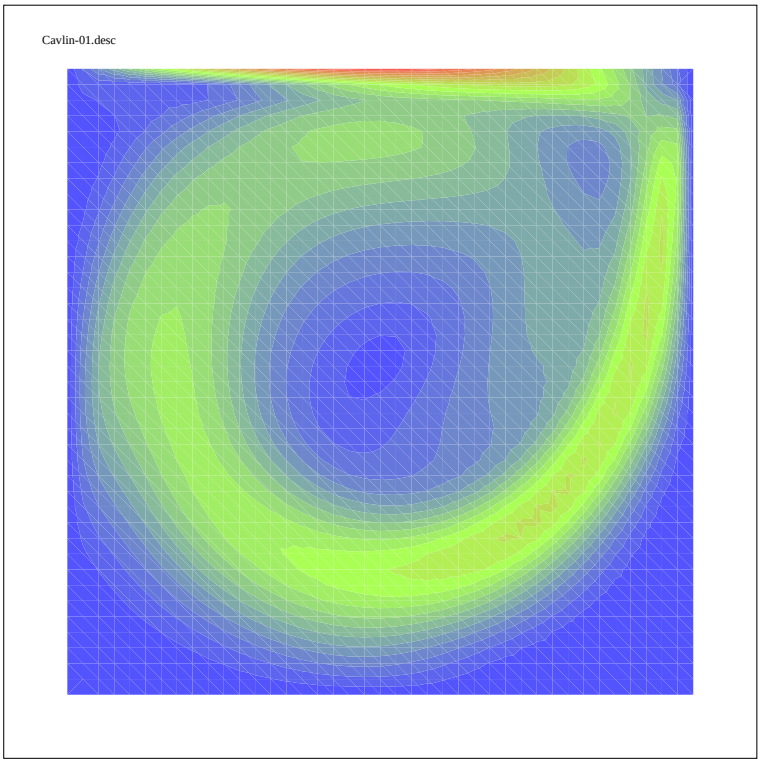

Figure 15: l_2 norm of the linearized driven cavity at steady state

produces a better approximation of the gradient compared to the first order scheme. This may be explained by the relatively large time step used for both schemes : $\delta\rho$, when small enough, fails to influence the topology of the fluid because of the numerical dissipation induced by the choice of large time steps. But, whereas this is not a surprise for the first order scheme, the second order scheme should be less diffusive. We suggest three reasons that could explain this fact :

- The numerical errors do not allow to reach a good agreement with the discretized gradient for $\delta\rho$ less than 10^{-6}. This was already observed for the Poiseuille flow. Thus, the second order scheme cannot achieve full performance.
- The derivative consistency was performed according to both first order and second order derivative cost function. This means that the first order (resp. second order) finite difference derivative is compared to the linearized derivative computed with the first order (resp. second order) scheme.

First Order LG		
Time Step	Cost	Cost grad.
0.2	0.648	1.028
0.1	0.766	1.313
0.05	0.859	1.604
0.01	1.026	2.091
Second Order LG		
Time Step	Cost	Cost grad.
0.2	1.011	2.010
0.1	1.043	2.204
0.05	1.013	2.168
0.01	1.152	2.601

Table 5: Cost function and cost function gradient for the first and second order Lagrange-Galerkin method

- For this test, the characteristic path computation was performed using the sub-cycling backtracking strategy. Unlike the adaptative backtracking, this algorithm requires that we provide a splitting parameter to our code. Unfortunately, the optimal parameter cannot be computed a priori and it may be easily under or overestimated.

We shall now develop the two last points.

First and second order comparison The table 5 gives the cost and cost function gradient for both first and second order LG schemes with several time steps. If we consider the first order LG scheme with $\Delta t = 0.001$ (Tab. 3) as our reference solution, several facts may be outlined :

- The first order Lagrange-Galerkin scheme seems to converge towards a limit as the time steps tends toward zero and therefore seems consistent in time.
- The second order Lagrange-Galerkin with $\Delta t = 0.2$ cost function and cost function gradient are already a very good approximation since it almost matches the reference first order LG scheme values.
- As Δt tends toward zero, the second order LG scheme behaves erratically.

The first explanation concerning the last point is that for the second order scheme, in some time steps range, the caracteristic feet reach the edges of some mesh elements, as the number of substeps is doubled compared to the first order scheme. However, this is unlikely, since the cost function is also subjected to an irregular behavior. The second explanation is based upon the conditional stability of the LG scheme. As stated before, when inexact integration of the right hand side is used, the Lagrange-Galerkin scheme suffers from unconditional

Second Order LG		
l_{split}	Cost	Cost grad.
5	1.012	2.017
10	1.011	2.010
20	1.010	2.003
30	1.010	2.003

Table 6: Cost function and its gradient for the second order Lagrange-Galerkin method with various splitting parameters

instabilities for some specific CFL numbers range (see [33] for instance). As a result, for this range of CFL numbers, oscillations may appear and produce numerical pollution inside the computation of the cost function and its gradient.
Finally, it is commonly admitted that the LG scheme is not efficient when using small time steps (see [6]) since numerical errors appear when the characteristics path is too short. This may explain the large values of the cost functions obtained for the second order LG scheme with $\Delta t = 0.01$. Since we are using the subcycling backtracking method in order to compute the characteristics path, the latter issue is crucial. The lack of a priori rule concerning the value of the splitting parameter leads us to perform a parameter sensitivity analysis in the next paragraph.

6.2.4 Influence of the splitting parameter

Problem statement As stated before, the algorithm used to compute both the steady state flow and the linearized flow was the subcycling backtracking method. This method, however more accurate than the initial method for large time steps, requires a splitting parameter which cannot be deduced a priori and may be easily under or over-estimated. If the splitting parameter is under-estimated, the characteristics path tracking procedure can produce a large number of cells jump at once, increasing by the way the diffusivity of the scheme.
On the other hand, over-estimating this parameter will drastically increase the characteristics time computation as well as numerical errors, since a characteristic speed needs to be interpolated at every sub step of the scheme. Whatsmore, large splitting parameters may also increase the probability to reach a mesh element boundary, thus ruining derivability since we are using a $\mathbb{P}_1$ finite element method.

Splitting parameter efficiency In order to illustrate the difficulties described above, we decide to check the gradient consistency using various splitting parameters. Fig. 16 and Tab. 7 give the gradient consistency graph and the computation time for the second order Lagrange-Galerkin method with $\Delta t = 0.2$ using various splitting parameters.

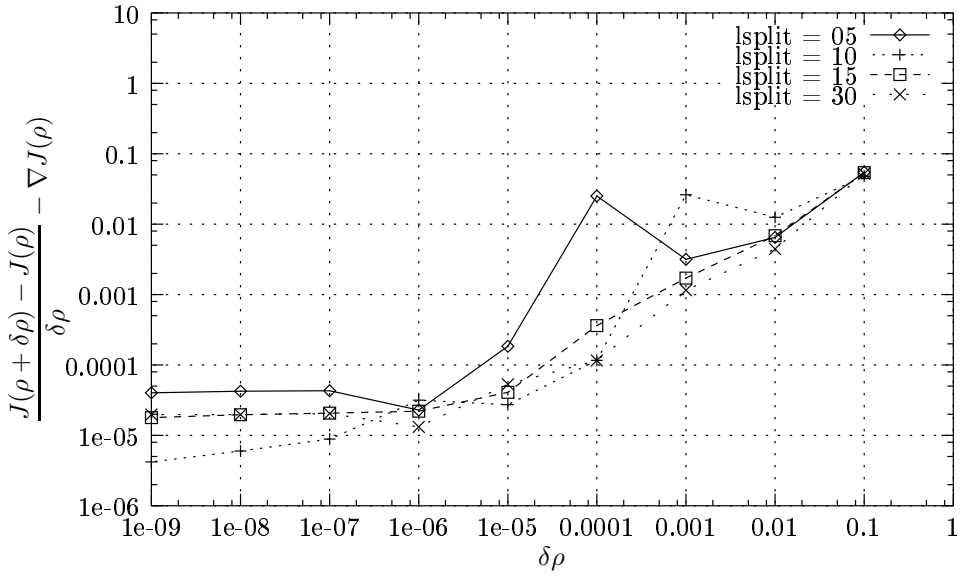

Figure 16: Splitting parameter efficiency for the second order LG scheme with $\Delta t = 0.2$

Second Order LG				
l_{split}	Tot. Time	Char. Time	%	Res. Time
5	200	89	44	90
10	300	156	52	91
20	360	246	68	89
30	459	347	75	90

Table 7: Computation time for the second order LG scheme with various splitting parameters ($\Delta t = 0.2$)

Second Order LG		
l_{split}	Cost	Cost grad.
01	1.015	2.192
05	1.013	2.171
10	1.013	2.168
20	1.014	2.169

Table 8: Cost function and its gradient for the second order Lagrange-Galerkin method with various splitting parameters ($\Delta t = 0.05$)

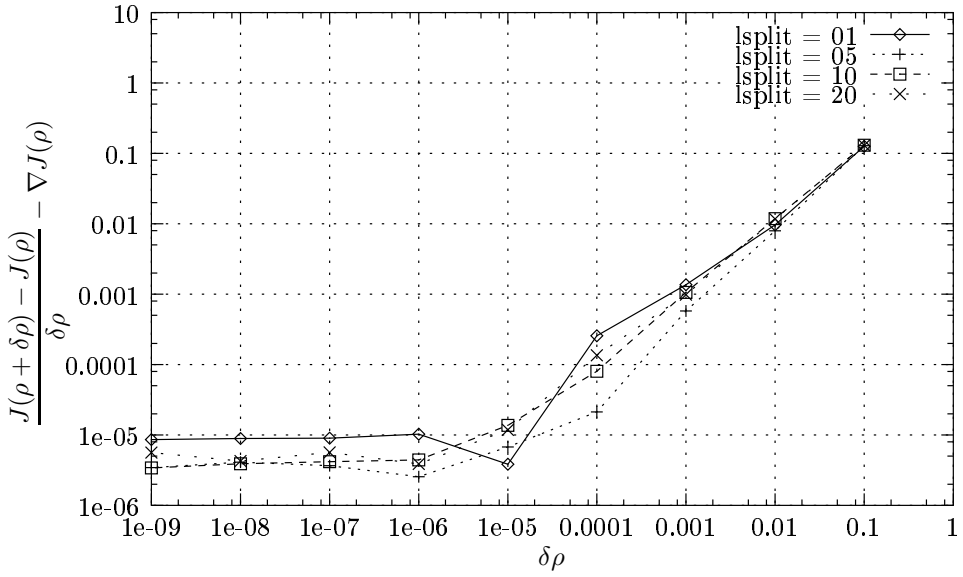

Figure 17: Splitting parameter efficiency for the second order LG scheme with $\Delta t = 0.05$

From Tab. 6 and Fig. 16, we can see that setting $l_{\text{split}} = 5$ gives the worst results, specially for $10^{-6} \leq \delta\rho \leq 10^{-3}$. On the contrary, $l_{\text{split}} = 20$ and $l_{\text{split}} = 30$ furnish similar values, slightly less accurate than the reference splitting parameter $l_{\text{split}} = 10$ for small $\delta\rho$ but show a significantly better behavior for $\delta\rho > 10^{-4}$.
Finally, our reference splitting parameter is clearly inaccurate for $\delta\rho > 10^{-3}$ but performs better below this value. Another side effect of increasing the splitting parameter is that it will automatically increase the computation time. If we look at Tab. 7, we clearly see the influence of a large splitting parameter over the overall computation time : while the resolution time remains the same, for $l_{\text{split}} > 20$ the characteristics path computation represents up to 75% of the total computation time, making the whole computation more than twice as expensive as $l_{\text{split}} = 5$ case. Hence, large spitting parameters should not be considered, at least for this time step. In Fig. 17, we ran the same test with a time step set to 0.05 and with various splitting parameter values. The cost function values are given in Tab. 8. As before, we can see that there seems to be an optimal value for l_{split}. But this time, as the time step is smaller, this optimal parameter seems to be close to 10: the costs values are almost identical for $l_{\text{split}} = 10$ and $l_{\text{split}} = 20$. Another obvious remark is that the case with no splitting does not produce good approximations since the cost function values are obviously too large for this range of time steps.

Second Order LG		
l_{split}	Cost	Cost grad.
10	1.152	2.601
01	1.153	2.603

Table 9: Cost function and its gradient for the second order Lagrange-Galerkin scheme with $\Delta t = 0.01$

Conclusion As a partial conclusion, we shall point out the following remarks,

- The subcycling backtracking method behaves well for the computation of the linearized state for the driven cavity. This is a good point since this flow features some non-linear aspects that the Poiseuille flow configuration does not take into account.
- The subcycling backtracking method requires an optimal splitting parameter. Unfortunately this parameter can not be determined a priori.
- the second order LG scheme is significantly a better approximation than the first order scheme for large time steps.

6.2.5 Second order LG scheme and integration rule

Motivations In Tab. 5, we saw that, for $\Delta t = 0.01$, the second order Lagrange-Galerkin scheme produces abnormal cost functions and gradients. But for all these tests, as we stated before, the computation was performed with a constant splitting parameter of 10. Of course, as we just stated in the latter paragraph, such a splitting parameter for such a small time step is likely to introduce numerical instabilities. Hence, we decided to perform the same test but with a splitting parameter $l_{\text{Split}} = 1$. the result is displayed in Tab. 9. There is no difference between the two tests, the cost function and its gradient being equal up to 10^{-3}. Since obviously the problem is not the splitting parameter, the quadrature rule should be the cause, as stated in the paragraph 6.2.3.

Integration rule influence The basic idea of the characteristics method is to turn the non-linear term $\frac{\partial}{\partial t} + u.\nabla$ operator expressed in Eulerian coordinates into a Lagrangian particular derivative $\frac{d}{dt}$. But this is possible only along the characteristics curves, defined by (10). Subsequently, when integrating the LG scheme over the time step Δt, one must provide the foot of the characteristics in order to integrate the right hand side. Unfortunately, this foot does not belong, in general, to the discrete space and therefore must be interpolated. This whole procedure results in an inexact integration of the right hand side by a quadrature integration rule, and the unconditional stability is lost. Even worse, ranges of time steps appear where the scheme is unstable, especially when the time step is small. But as stated

First Order LG		
Gauss points	Cost	Cost grad.
07 pts	1.039	2.135
12 pts	1.004	2.005
Second Order LG		
Gauss points	Cost	Cost grad
07 pts	1.153	2.603
12 pts	1.062	2.133

Table 10: Cost and cost function gradient for the second order Lagrange-Galerkin scheme with $\Delta t = 0.01$ for both 7 and 12 points quadrature

first Order LG			
Gauss points	Tot. time	Char. time	%
07 pts	2959	1455	32
12 pts	5236	2610	50
Second Order LG			
Gauss points	Tot. time	Char. time	%
07 pts	5251	2847	54
12 pts	7477	5094	68

Table 11: Computation time for the first and second order Lagrange-Galerkin scheme with $\Delta t = 0.01$ for both 7 and 12 points quadrature

in [33] and [15], increasing the number of integration points reduces these instabilities. So far, the integration rule we used was a 7-point Gauss rule. Let us now consider the same test, but with a 12-point Gauss rule. Tab. 10 gives the cost function and its gradient for $\Delta t = 0.01$. The splitting parameter was set to $l_{split} = 1$.

Clearly, the use of a more accurate integration quadrature reduces the instabilities of the second order Lagrange-Galerkin in conjunction with small time steps. However, the counterpart is an increase of the computation time as displayed in Tab. 11 : the characteristics computation time is almost doubled for both first and second order algorithm.

Nevertheless, when a 12 Gauss point integration is used, the cost values seem to be more within the range of each LG scheme order. We can therefore conclude that the more Gauss point you use, the better the approximation is. This may not be entirely true since rising the number of integration points also increases the number of characteristics paths to be computed, hence rising the odds to reach the edge of an element where the NS equations are not derivable for $\mathbb{P}_1$ finite element class.

Conclusion For that reason, we may conclude as follow:

- The second order LG scheme is very unstable when small time steps are used.
- Making the second order scheme stable requires that we use more Gauss point for the right hand side integration, making the overall algorithm very expensive.
- The first order LG scheme is less unstable than the second order scheme for small time steps.
- For small time steps, both first order and second order schemes when stabilized perform almost the same way.

7 Drag reduction around a rotating cylinder

In this section, we shall apply our optimal control strategy in order to reduce the drag induced by a fluid flow around a rotating cylinder. The control parameter is the angular velocity of the cylinder that will be considered either constant or time-dependent. This problem has been investigated previously by many authors, on the experimental and numerical point of view, we refer to [26] and [29] for a complete review on this subject. Our goal is not to improve earlier results, but try to validate our optimization strategy based on the Lagrange-Galerkin scheme for the Navier-Stokes system and a quasi-Newtonian method for the optimization procedure.

7.1 Control problem setting

We consider an incompressible Navier-Stokes flow around a circular cylinder of radius $a = 0.5$ (see Fig. 18). The fluid is assumed to be a viscous incompressible newtonian fluid. Its evolution is described by its velocity u and its pressure p. The couple (u, p) satisfies the classical Navier-Stokes equations written in non-conservative form :

$$\begin{cases} \partial_t u + \nabla u \cdot u - \nu \Delta u + \nabla p = 0, & \Omega \times (0,T) \\ \mathrm{div}(u) = 0, & \Omega \times (0,T) \\ u = g, & \Gamma_{cyl} \times (0,T) \\ \sigma(u,p) \cdot n = 0, & \Gamma_{out} \times (0,T) \\ u = u_\infty, & \Gamma_{in} \times (0,T) \\ u(t=0) = u_0, & \Omega \end{cases} \tag{70}$$

where ν stands for the kinematic viscosity and u_∞ is the farfield velocity field. The quantity $\sigma(u,p) = -p\,\mathrm{I} + \nu(\nabla u + \nabla^T u)$ stands for the fluid stress tensor inside Ω, with $(\nabla u)_{ij} = \partial_j u_i = u_{i,j}$.
We assume the velocity g to be tangent to the cylindar, we define the tangent vector to the cylinder,

$$\begin{cases} \tau_1(x,y) &= -y, \\ \tau_2(x,y) &= x, \end{cases} \tag{71}$$

and we set $g = \omega(t)\,\tau$ with $\omega(t)$ stands for the amplitude of the angular velocity of the cylinder.
Hence the parameter of control is here $\omega(.) \in \mathcal{U}_c$, where $\mathcal{U}_c$ will be described later on.
We define the Reynolds number as follows,

$$Re = \frac{2a|u_\infty|}{\mu} \tag{72}$$

We set $|u_\infty| = 1$ and $\mu = \frac{1}{200}$. Hence we are working with a Reynolds number $Re = 200$.
The Strouhal number S_a is defined by

$$S_a = \frac{2af}{|u_\infty|} \tag{73}$$

where f stands for the frequency of the vortex shedding.
We would like to solve the following minimization problem :

$$\min_{\omega \in \mathcal{U}_{ad}} J(u(\omega), p(\omega)) \tag{74}$$

where (u, p) satisfies the Navier-Stokes system and the cost function J(u,p) measures the mean square of the drag around the cylinder, i.e

$$J(u,p) = \int_0^T \left| \left(\int_{\Gamma_{Cyl}} \sigma(u,p) \cdot n \, d\Gamma \right) \cdot e_1 \right|^2 dt \tag{75}$$

In the sequel, we will choose different admissible control spaces $\mathcal{U}_c$ of finite dimension.

7.2 Case of a single harmonic angular velocity

7.2.1 Continuous optimal control

We set $\omega(t) = \rho \cdot \sin(2\pi S_e t)$. The quantities (ρ, S_e) stand for the forcing amplitude and frequency parameters. This means that we set,

$$\left\{ \begin{array}{rcl} g_1(x,y,t) & = & -\rho \cdot \sin(2\pi S_e t)\, y \\ g_2(x,y,t) & = & \rho \cdot \sin(2\pi S_e t)\, x \end{array} \right. \tag{76}$$

We are interested in solving the following minimization problem,

$$\min_{(\rho, S_e) \in \mathbb{R}^2} j(\rho, S_e) \tag{77}$$

where

$$j(\rho, S_e) = J(u(\rho, S_e), p(\rho, S_e)) \stackrel{\text{def}}{=} \int_0^T \left| \left(\int_{\Gamma_{Cyl}} \sigma(u(\rho, S_e), p(\rho, S_e)) \cdot n \, d\Gamma \right) \cdot e_1 \right|^2 dt \tag{78}$$

In order, to perform an optimization procedure as described in section (1), we need to compute the gradient of the cost function $j(\rho, S_e)$ with respect to the design variables (ρ, S_e). This gives,

$$
\begin{aligned}
&< \nabla j(\rho, S_e), (\delta\rho, \delta S_e) > \\
&= 2 \cdot \int_0^T \left[\left(\int_{\Gamma_{Cyl}} \sigma((u,p)(\rho, S_e)) \cdot n \, d\Gamma \right) \cdot e_1 \right] \cdot \left[\left(\int_{\Gamma_{Cyl}} \sigma((z,q)(\delta\rho, \delta S_e)) \cdot n \, d\Gamma \right) \cdot e_1 \right]
\end{aligned}
$$

where $(z,q)(\delta\rho, \delta S_e) = \frac{Du}{D(\rho,S_e)}(\rho, S_e) \cdot (\delta\rho, \delta S_e)$ stands for the state derivative with respect to the parameters (ρ, S_e) in the perturbation direction $(\delta\rho, \delta S_e)$ and is solution of the following linearized system,

$$
\left\{
\begin{array}{ll}
\partial_t z + \nabla u \cdot z + \nabla z \cdot u - \nu \Delta z + \nabla q = 0, & \Omega \\
\mathrm{div}(z) = 0, & \Omega \\
z = \left\{ \begin{array}{l} \delta\rho \sin(2\pi S_c t)\, \tau \\ 2\pi\rho\delta S_e \cos(2\pi S_c t)\tau \end{array} \right. , & \Gamma_{cyl} \\
\sigma(z,q) \cdot n = 0, & \Gamma_{out} \\
z = 0, & \Gamma_{in} \\
z(t=0) = 0, & \Omega
\end{array}
\right. \tag{79}
$$

7.2.2 Numerical experiments

At that point, we apply the numerical strategy described in section (4) with the parameters, in the following table

Time	Space
Characteristic of order 2	$\left(\mathbb{P}^1 \oplus \mathbb{P}^b(K)\right)^2 \times \mathbb{P}^1$
$\mathrm{dt} = 5.10^{-2}$	$\mathrm{meas}(K_h) \in [10^{-3}, 10^{-1}]$
7 Gauss points	1363 vertices 2560 triangles

We have first performed a direct parameter analysis, in order to characterize the shape of the cost function and track some local optimum. The Fig. 19 and Fig. 20 show the contour

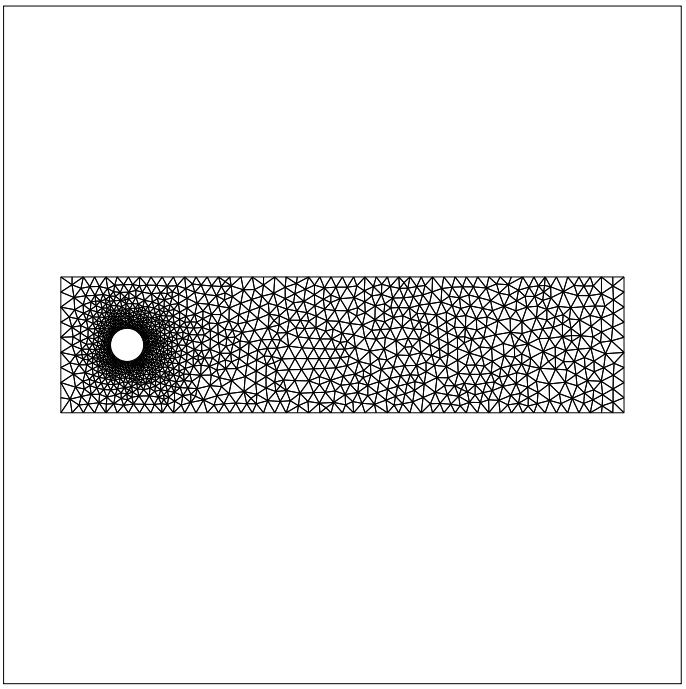

Figure 18: Computational mesh around the cylinder

plot of the cost function. It may be emphasized that the functional is apparently convex and coercive, since its shape is almost close to a parabola. This means, that the optimization procedure should fast converge to the optimum point, which seems to be unique, at least in the range of our direct sensitivity analysis.
We found by this simple approach a minimum value at point ($\rho = 4, S_e = 0.9$). We call this point the direct optimal value. We have plotted in Fig. 21 and Fig. 22, the drag and the lift evolution for three different sets of parameters: the uncontrolled case parameters, the case of arbitrary parameters and the one for the direct optimal parameters.
We can see that our control affects efficiently the drag history, with a rapid decrease of the instantaneous drag value. On the contrary, the lift coeficient is much less perturbed by the rotation of the cylinder. Only its frequency is affected by the forcing rotation. This can be explained by the fact that the mean flow is horizontal and the rotation of the cylinder does not really perturb the transverse flow.
Near the direct optimal parameters, we started an optimization procedure using the quasi-Newton BFGS algorithm described in section 4. We found an optimal value at point ($\rho = 3.986, S_e = 0.942$) after 15 iterations with a small decrease of 0.15% compared to the direct optimal value. The overall drag reduction compared to the uncontrolled case represents

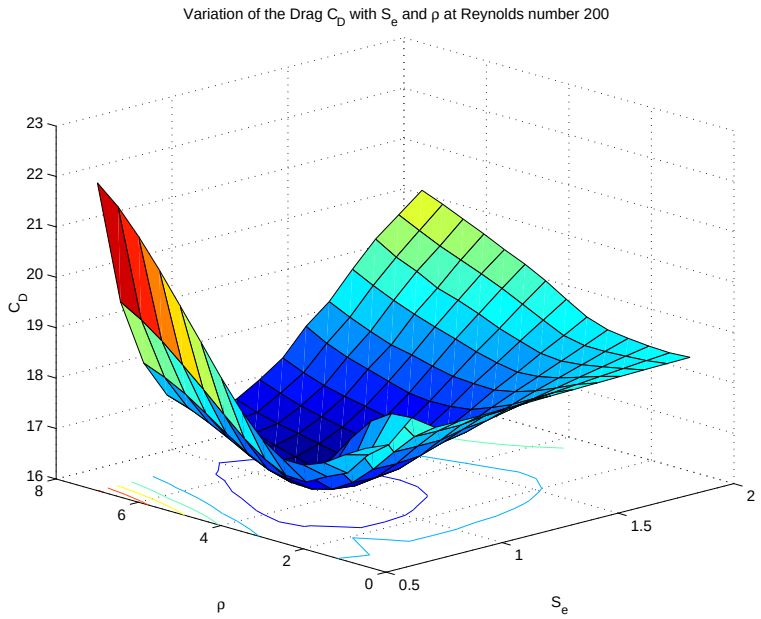

Figure 19: 3D contour plot of the cost function $j(\rho, S_e)$

11.5% of the uncontrolled mean drag.
We have represented the flow topology for the uncontrolled case and the optimal controlled case at a given time $t = 10$ in Fig. 23 and Fig. 24. We see that the control effect is to suppress or at least weaken the vortex shedding phenomenon. This should have been predicted, since the mean source of friction forces is due to the vortex shedding alley as reported by many experiments [26].

7.3 Case of several harmonics

In order to enrich the control parameter space, we chose to deal with a control which is variable as an harmonic series based on the forcing Strouhal number $S_{opt} = 0.942$ found in the previous section, i.e

$$\omega(t) = \sum_{k=1}^{N} \rho_k \cdot \sin(2k\pi S_{opt} t)$$

as reported in [26]. Hence, the control parameters are the family $\{(\rho_k)\}_{1 \leq k \leq N}$. We perform the optimization procedure for $N = 3$. We found that the effect of higher harmonics is negligible as shown in Fig. 25. But still our algorithm performs well and found an optimal value for the three parameters. The computations ran on three computers with the same clock

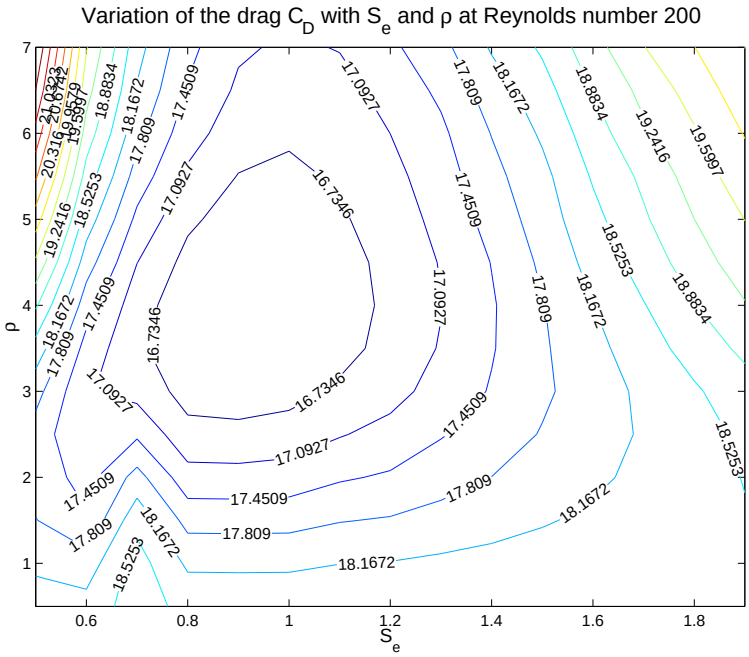

Figure 20: 2D contour plot of the cost function $j(\rho, S_e)$

speed, leading to an equivalent cpu time compared to the case with only two parameters. We finally, recall the mean drag value for the different optimization procedures,

Case of study	Mean Drag C_D
uncontrolled	0.92576
direct optimal control	0.8188215
single harmonic optimal control	0.8186945
3 harmonics optimal control	0.817156

8 Identification of far-field boundary conditions from fluid loads on bluff bodies

In this section, we are interested in an identification problem arising in the aeroelastic stability analysis of elastic structures inside a fluid flow [34]. The goal is to identify inflow velocity fields from the knownledge of fluid loads time history on a fixed solid that may represent the 2d-section of a bridge deck [39, 2].
The extension of this work to fluid-structure interaction context may be appropriate to

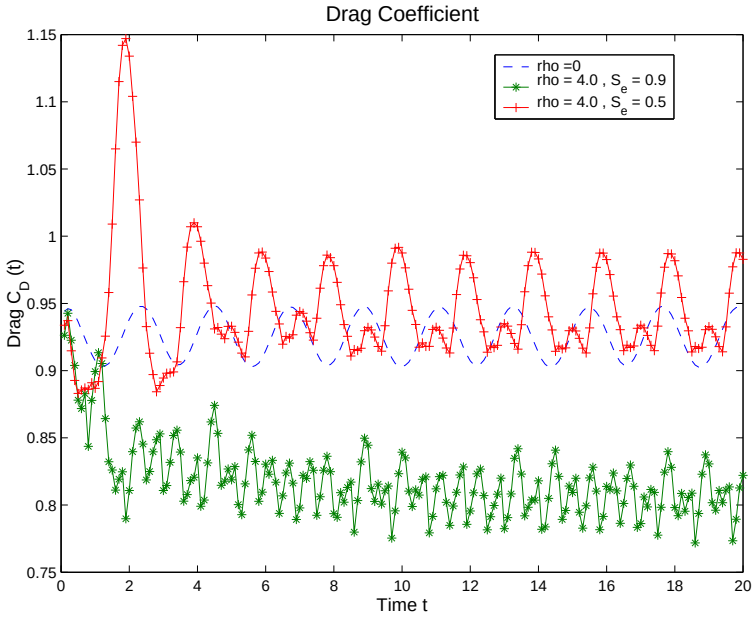

Figure 21: Drag evolution for different control parameters

identify the inflow velocity field inducing unstable structural displacements. This work is a first basic attempt towards the full coupled case and that needs several improvements in order to reach efficiency.

8.1 Problem settings

We consider an incompressible Navier-Stokes flow around a rectangular bluff body $\Omega^s \in \mathbb{R}^2$. The couple (u, p) satisfies the classical Navier-Stokes equations,

$$\begin{cases} \partial_t u + \nabla u \cdot u - \nu \Delta u + \nabla p = 0, & \Omega \times (0,T) \\ \mathrm{div}(u) = 0, & \Omega \times (0,T) \\ u = 0, & \Gamma_s \times (0,T) \\ \sigma(u,p) \cdot n = 0, & \Gamma_{out} \times (0,T) \\ u = u_\infty, & \Gamma_{in} \times (0,T) \\ u(t=0) = u_0, & \Omega \end{cases} \tag{80}$$

This model is appropriate in order to analyse aerodynamic characteristics of structures inside atmospheric fluid flows. The solution of such a system has been computed using the discretization described in section 3.
Our main concern is to identify the inflow boundary velocity u_∞ on Γ_{in} which produces a

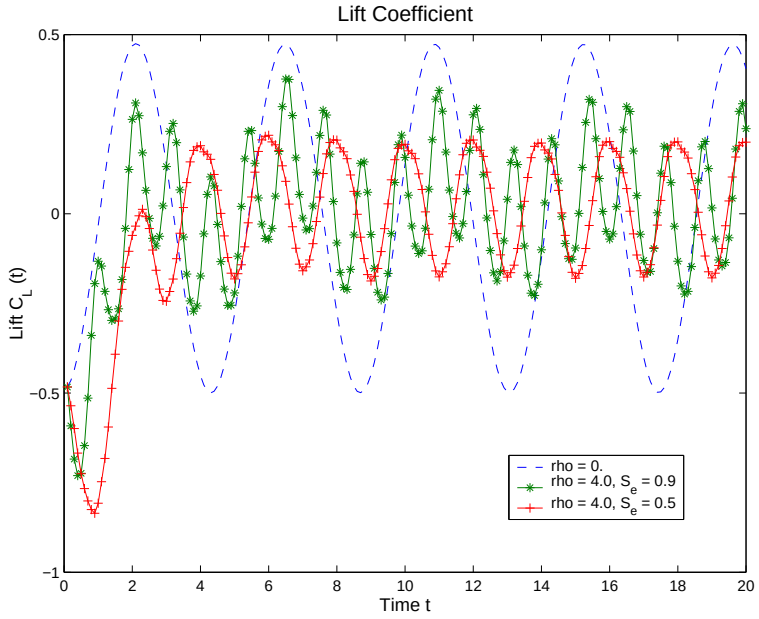

Figure 22: Lift evolution for different control parameters

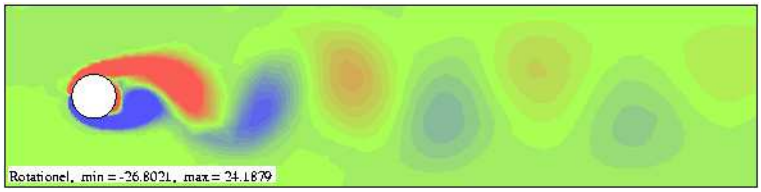

Figure 23: Vortex shedding in the uncontrolled case

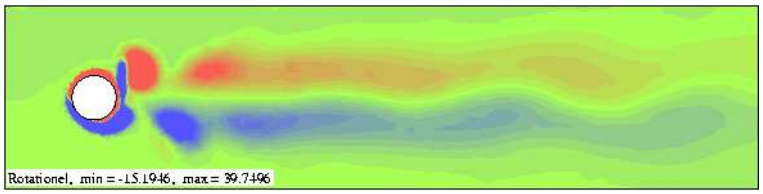

Figure 24: Vortex shedding in the optimal controlled case

fluid load on Γ^s arbitrary close to a given load target $F_d(t)$ on the time intervall $(0,T)$. This

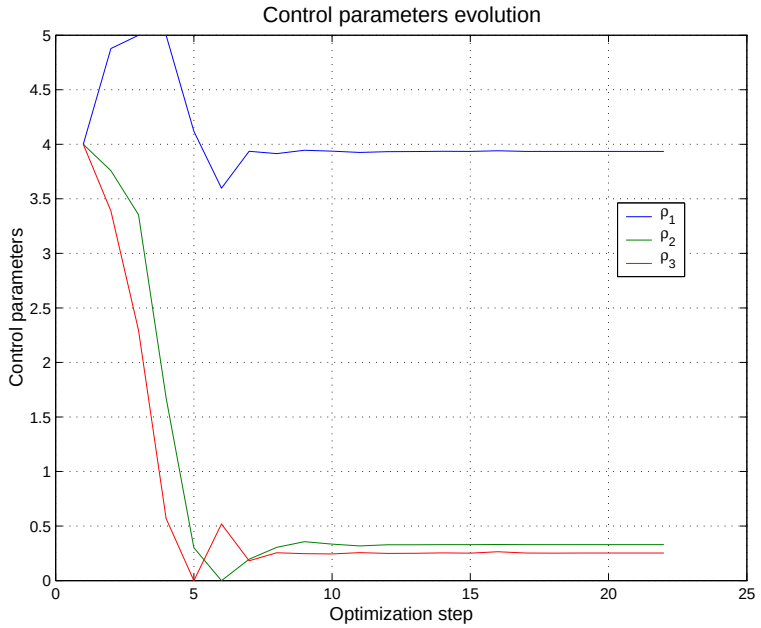

Figure 25: Optimization parameters evolution

can be solved by using the following minimization problem,

$$\min_{u_\infty \in \mathcal{U}_{ad}} J(u(u_\infty), p(u_\infty)) \tag{81}$$

where (u, p) satisfies the Navier-Stokes system. Here, we only consider the fluid load corresponding to the mean flow direction, i.e the drag. Then, the cost function $J(u, p)$ measures the mean square of the difference between the current drag and the target drag time history,

$$J(u,p) = \int_0^T \left| \left(\int_{\Gamma^s} \sigma(u,p) \cdot n \, d\Gamma \right) \cdot e_1 - F_d(t) \right|^2 dt \tag{82}$$

Using exactly the same framework as for optimal control of rotating cylinder drag, it is possible to derive the gradient of the cost function $j(u_\infty)$ with respect to the inflow velocity in the perturbation direction δu_∞.

8.2 The rectangular cylinder

Because of its profiled shape, the cylinder used for our tests in the previous section is of little interest as far as fluid-structure interaction is concerned. Its curved shape was barely a challenge for the Lagrange-Galerkin method. We shall this time consider the rectangular

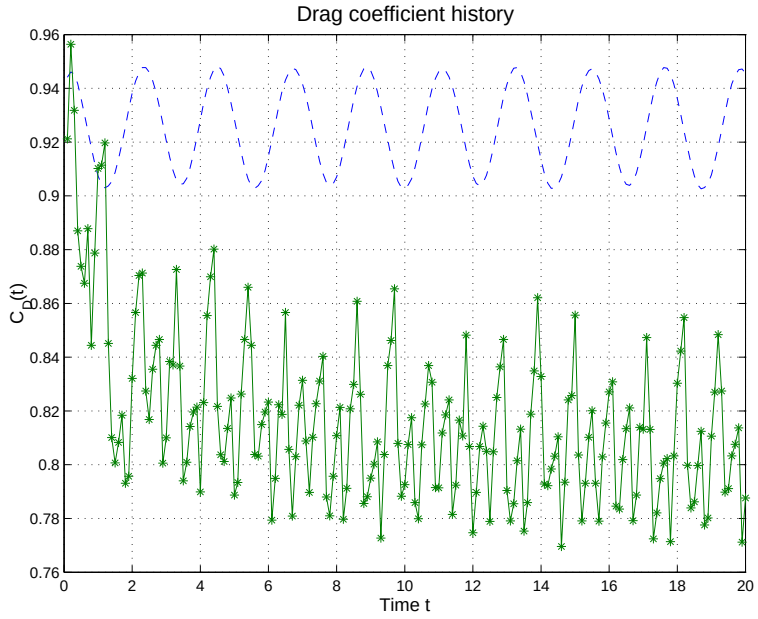

Figure 26: Optimal Drag time evolution - 3 harmonics case

cylinder. This test is popular in wind-engineering because, though geometrically very simple, the right angles it provides are very representative of the object shapes commonly studied such as bridges or towers for instance. The rectangular cylinder we will consider consists of a rectangle with chord to thickness ratio $c/d = 4$, where c is the chord and d the length. Fig. 29 and Fig. 30 give the lift, drag and the pressure profile at $t = 500$ seconds for the second order Lagrange-Galerkin scheme with $Re = 1000$ computed with the code LGNSI2FS. Like the cylinder, we can see that after a transient phase the perturbation of the flow becomes periodic. Again, the lift oscillation frequency f_l is half the drag oscillation frequency f_d and the corresponding Strouhal number here is $S = f_l d/u_\infty = 0.105$.

8.3 Harmonic perturbation of the inflow velocity

As in the previous section, since we do not use an adjoint formulation, we have to restrict drastically the class of admissible controls. Since we have in mind the case where the solid is moving using a single second-order differential equation of spring-mass type, we use an harmonic perturbation around a steady state for the inflow velocity,

$$u_\infty(\rho, S_e, \phi) = 1 + \rho \sin(2\pi S_e t + \phi) \tag{83}$$

This means that we must use a control space of dimension 3. Of course this kind of controls restricts the kind of load time history that may be reached. But since the basic flow while

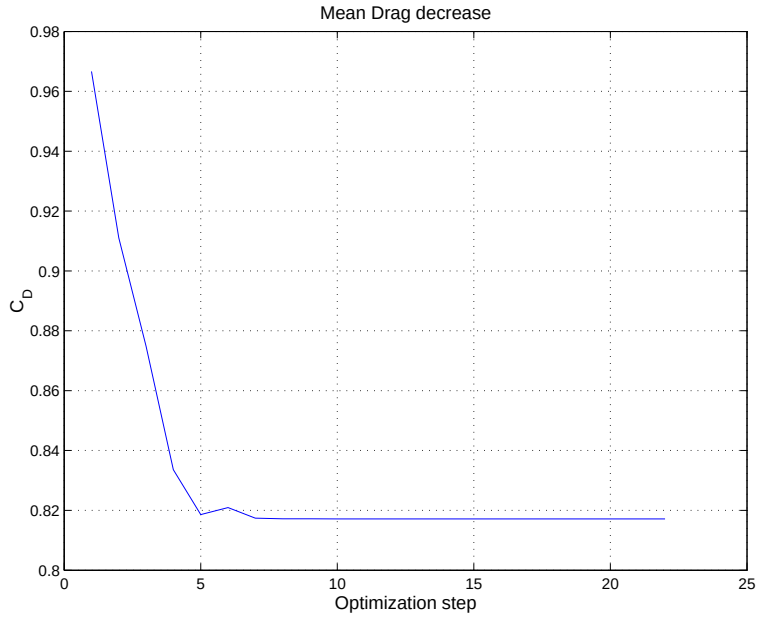

Figure 27: Mean drag evolution- 3 harmonics case

using steady inflow velocity produces harmonic fluid loads on Γ^s, this choice is admissible. We will first use a synthetic load F_d produced by a set of input control parameters and try to recover this set using an optimization procedure based on a Quasi-Newton gradient based method. In a second time, we will define an harmonic arbitrary load and track this load using optimal control parameters.

8.4 Synthetic load

In this paragraph, we shall use a reachable target, i.e that has been obtained by selecting a set of control parameters and by computing the result drag time history $F_d(t)$ on the R4 profile. The target has been obtained using the inflow velocity $u_\infty^d = 1 + \rho^d \sin(2\pi S_e^d t)$ with $(\rho^d, S_e^d) = (0.3, 0.3)$.
We have displayed in Fig. 31, the control parameters during the optimization procedure, with an inflow velocity $u_\infty = 1 + \rho \sin(2\pi S_e t)$. The initial guess was set to $(\rho^0, S_e^0) = (0.5, 0.5)$. Convergence was reached after 12 iterations. We recover the awaited control parameter without any regularization process, which is a characteristic of the reachable target class. We did not deal with the question of the stability of this inverse problem, since our goal was to validate the computation of the cost function gradient.
We have used an harmonic inflow velocity in the direction of the drag, and we recover in Fig.

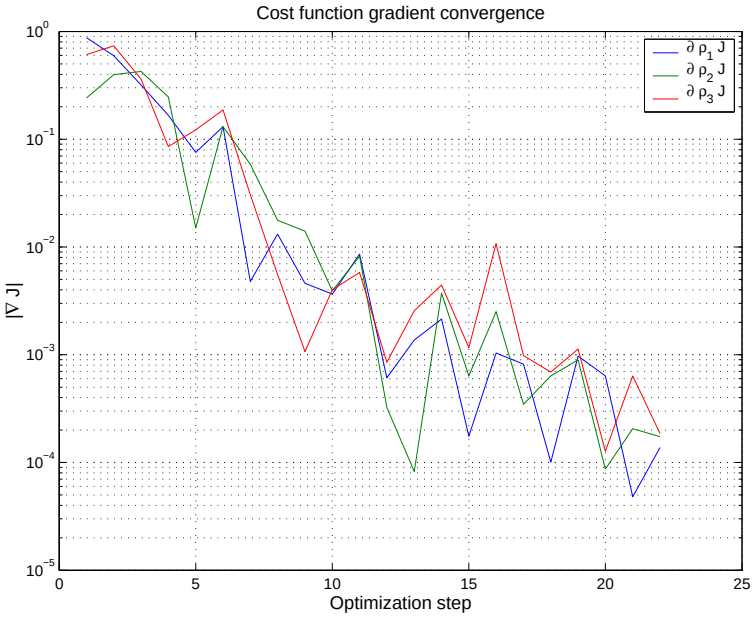

Figure 28: Cost function gradient decrease - 3 harmonics case

32 the frequency of the inflow inside the drag time history. This means that the frequency of the inflow strongly controls the frequency of the resulting drag. On the contrary, the frequency of the lift does not depend on the frequency of the inflow as shown in Fig. 34. This is due to the fact that the inflow velocity is horizontal and that the transverse fluid flow is not perturbed by the time variation of the mean horizontal flow. This is why we did not try to track the lift time history.

8.5 Arbitrary harmonic loads

The next step is to track an arbitrary fluid load. We mean here that this load does not result from fluid flow computations. Nevertheless, we cannot choose a completely arbitrary load, since the class of the inflow velocity strongly pilots the kind of load to be produced by the numerical simulations. For this reason, we choose an harmonic load oscillating at the fundamental frequency f^d,

$$F_d(t) = A^d \sin(2\pi f^d t) \tag{84}$$

with $(A^d, f^d) = (0.2, 0.5)$. We work with a inflow control integrating phase control,

$$u_\infty = 1 + \rho \sin(2\pi S_e\, t + \phi) \tag{85}$$

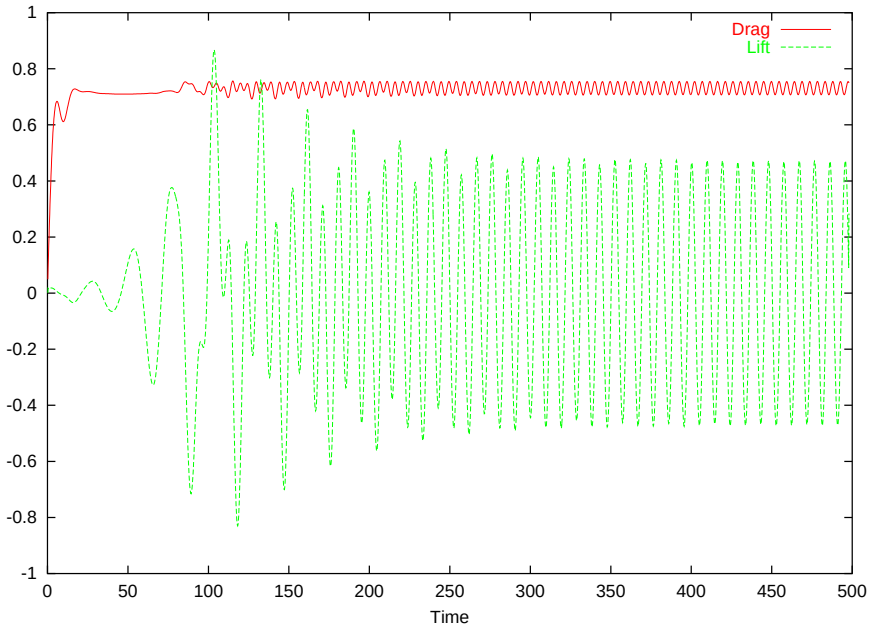

Figure 29: Lift and drag for the R4 rectangular cylinder

Initial time condition for the fluid is set with a developed flow obtained at time $t = 20s$ with an inflow velocity $u_\infty = 1$. The optimization process was performed with the initial guess $(\rho^0, S_e^0, \phi) = (0.5, 0.4, 0)$. In Fig. 35, we have displayed the control parameters during the optimization process. Convergence was reached after 31 iterations of the L-BFGS-B algorithm on 3 different processors. The tracking functional was highly reduced as shown in Fig. (36). The different partial derivatives during the optimization are displayed in Fig. 37. Compared to the case of a synthetic load, the convergence was slower and the reduction factor was weaker. This is due to the fact that the coercivity of the functional is not ensured in this case, leading to poor conditioning property of the linear systems involved in the computation of the descent directions. Nevertheless, convergence was reached, what might indicate that our linearized algorithm behaves well even in non trivial situations.
Furthermore, these numerical experiments may shed some light about the identifiability question for the Navier-Stokes system. Indeed, it has been possible to identify the Dirichlet boundary condition on Γ_∞^f from the knowledge of the mean of the normal stress tensor on Γ^s. This means that the approximate identification property is certainly true, even if it has not been proven yet. We note that for the Stokes system, a similar identifiability property was proven theoretically in [34].

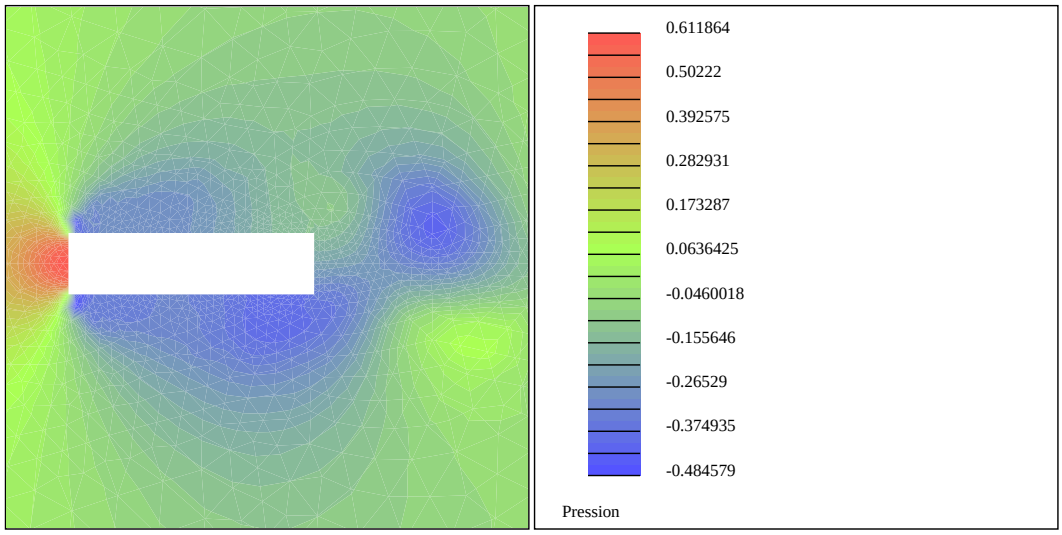

Figure 30: Pressure profile for the R4 rectangular cylinder

9 Identification of far-field boundary conditions from fluid loads on moving bluff bodies

In this section, we may extend the setting of the previous section to moving domain configuration. The goal remains the identification of fluid velocities on the far-field boundary Γ_∞^f thanks to the knowledge of fluid loads on a moving profile inside the fluid flow.

9.1 Problem settings

We consider an incompressible Navier-Stokes flow around a moving rectangular bluff body $\Omega^s \in \mathbb{R}^2$. The couple (u, p) satisfies the ALE Navier-Stokes system ,

$$\left\{\begin{array}{ll} \frac{\partial u}{\partial t}_{|\xi\in\Omega^f(\tau)} + \nabla u \cdot (u - w^\tau) - \nu\Delta u + \nabla p, & \Omega_t^f \\ \mathrm{div}(u) = 0, & \Omega_t^f \\ u = \dot{x}^s, & \Gamma_t^s \\ \sigma(u,p)\cdot n = 0, & \Gamma_{out} \\ u = u_\infty, & \Gamma_{in} \\ u(t=0) = u_0, & \Omega \end{array}\right. \tag{86}$$

The motion of the solid is rigid and prescribed as follows,

$$x^s = \left[\begin{array}{cc} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{array}\right] \cdot \left[\begin{array}{c} x \\ y \end{array}\right], \quad (x,y) \in \Gamma_0^s$$

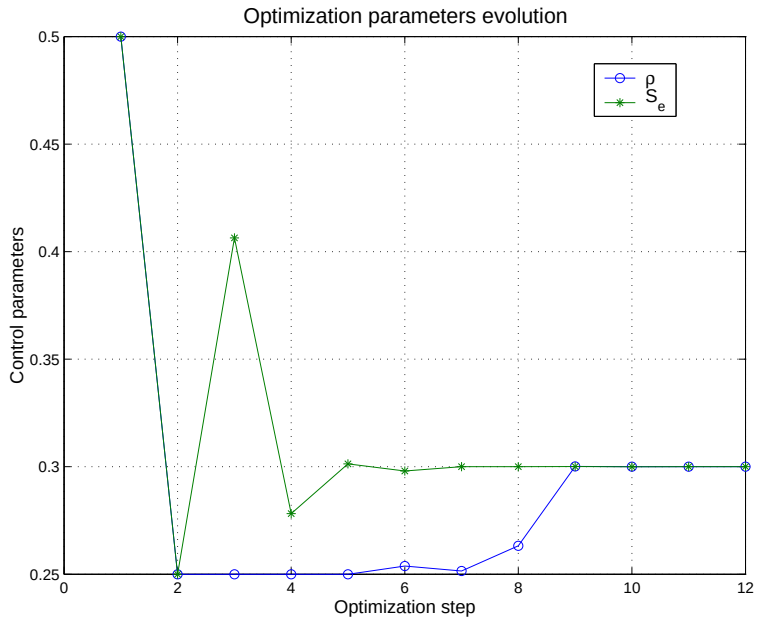

Figure 31: Control parameters during optimization steps

As before, we would like to identify the inflow boundary velocity u_∞ on Γ_{in} which produces a fluid load on Γ_t^s arbitrary close to a given load target $F_d(t)$ on the time intervall $(0, T)$. Hence we need to solve the following minimization problem,

$$\min_{u_\infty \in \mathcal{U}_{ad}} J(u(u_\infty), p(u_\infty)) \tag{87}$$

where $(u(u_\infty), p(u_\infty))$ satisfies the Navier-Stokes system and the cost function $J(u, p)$ measures the mean square of the drag difference,

$$J(u,p) = \int_0^T \left| \left(\int_{\Gamma_t^s} \sigma(u,p) \cdot n \, d\Gamma \right) \cdot e_1 - F_d(t) \right|^2 dt \tag{88}$$

9.2 Mesh movement algorithm

In the previous section, our computation domain Ω_h^s was fixed. Since we would like to control the fluid load history with fluid/structure interaction, we should be able to move the computation domain with respect to the fluid/structure interface [38]. Therefore, the reference domain Ω^s will evolve in time, following the structural displacement. The computational domain Ω_h^s will have to be updated in order to stick to $\Omega(t)$. Here, we assume we know

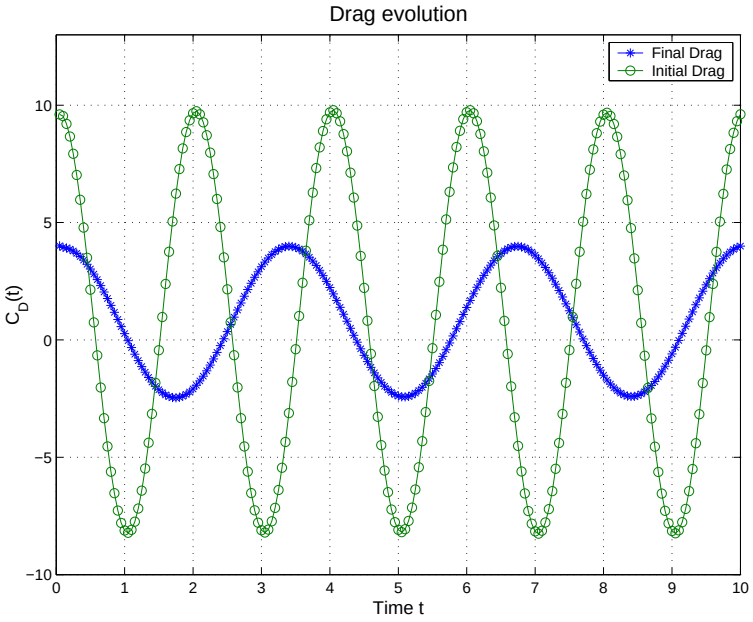

Figure 32: Drag history - initial guess and final control

the position of the fluid mesh Ω^s at $t = t^1$ and the position of the fluid/structure interface $\Gamma(t)$ at $t = t^2$. We will restrict ourselves to the case of a forced sinusoidal oscillation for the structure. The displacement angle $\theta(t)$ is given by the following formula:

$$\theta(t) = \theta_m \sin(\omega_m t) \tag{89}$$

The general algorithm used to update the fluid mesh is then the following:

- the far-field points Γ^h_∞ are fixed and therefore not updated.
- $\Gamma^s(t^2)$ needs to be computed. This will be done by using (89). For exemple, this can be done by simply applying this formulae to the orignal $\Gamma^s(t = 0)$, or computing the rotation between t^1 and t^2 using $\Gamma^h(t^1)$.
- The mesh points inside $\Omega^s \setminus \Gamma$ are updated using the following method: Each edge ij of the triangulation, i and j being the correponding vertices, is given a stiffness coeficient κ_{ij}. This coefficient is, for instance, the inverse of their length, i.e:

$$\kappa_{ij} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}$$

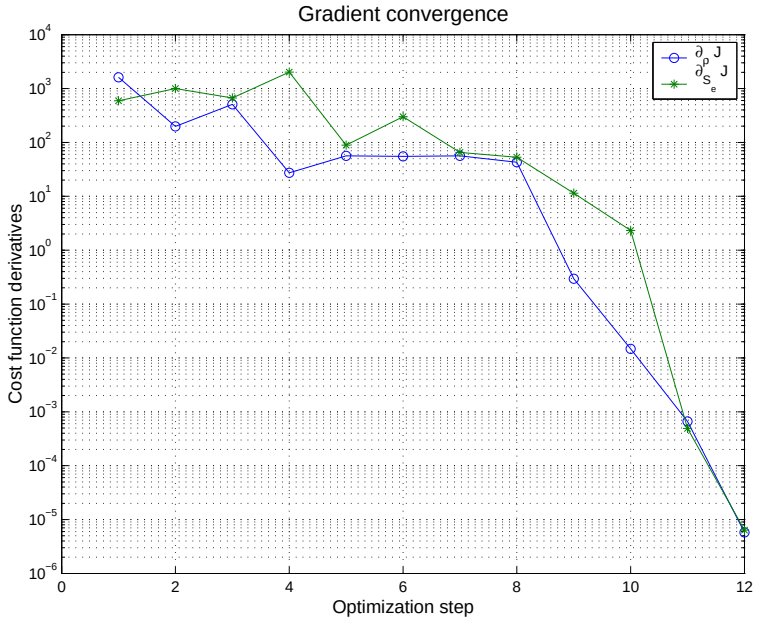

Figure 33: Gradient decrease during optimization steps

The goal is now to compute the displacement δ_i for all vertices i of the mesh Ω^s between t^1 and t^2. The algorithm we used is a Jacobi based iterative method. Given a prediction δ_i^p of the displacement, Jacobi iterations are performed, namely :

$$\delta_i^{n+1} = \frac{\sum_{j \in N(i)} \kappa_{ij} \delta_j^n}{\sum_{j \in N(i)} \kappa_{ij}} \tag{90}$$

for $i \in \Omega^s \setminus (\Gamma_h \bigcup \Gamma_\infty)$, and where $N(i)$ are neighbour vertices of i. This iterative procedure is initialised by the following set of data:

$$\begin{cases} \delta_i^0 &= 0, & \forall i \in \Gamma_\infty \\ \delta_i^0 &= x_i(t^2) - x_i(t^1), & \forall i \in \Gamma_h \\ \delta_i^0 &= \delta_i^p, & \forall i \in \Omega_s \end{cases}$$

After N iterations, when the residual between two successive iterations is under a given tolerance parameter, we get the displacement needed to update the mesh :

$$x_i(t^2) = x_i(t^1) + \delta_i^N, \qquad \forall i \in \Omega^s - \Gamma.$$

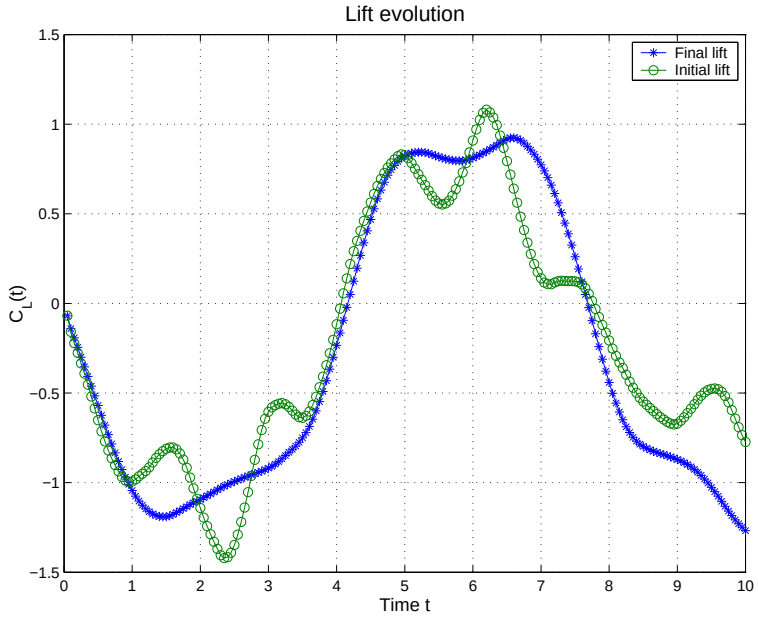

Figure 34: Lift history - initial guess and final control

Fig. (38) give the mesh vertices position for $t = 0s$ and $t = 16s$ using this procedure, with $\theta_m = 1$ deg. and $\omega_m = \frac{2\pi}{5}$.

Remark 16 *The convergence of the Jacobi iterative method is assured because the matrix associated to (90) is diagonal dominant.*

9.3 Discrete gradient consistency

For the time being, we have been analysing optimal control problems in fixed domains. Here the domain is moving with a given displacement of the solid boundary Γ_t^s. Since this situation is new, we have been checking the consistency of the discrete gradient computed by the LGNI2FS code.
First let us describe the linearized system using the subcycling backtracking procedure. We recall that the Navier-Stokes system is discretized as follows, we look for $(u_h^{n+1}, p_h^{n+1}) \in$

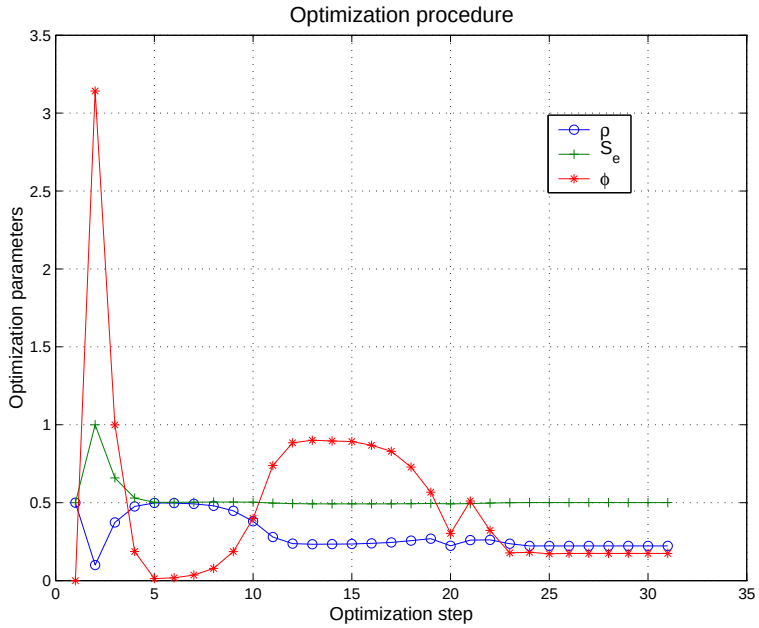

Figure 35: Control parameters during optimization steps

$X_h^3 \times Q^h$ solution of the following system,

$$\begin{cases} \frac{1}{\Delta t}(u_h^{n+1}, v_h) + \nu(\nabla u_h^{n+1}, \nabla v_h) - (p_h^{n+1}, \operatorname{div} v_h) = \frac{1}{\Delta t}(u_h^n \circ \chi_h^{u-w,n}, v_h), & \forall\ v_h \in V_h^{n+1} \\ (\operatorname{div} u_h^{n+1}, q_h) = 0, & \forall\ q_h \in Q_h^{n+1} \\ u_h^{n+1} = (\dot{x}^s)_h^{n+1}, & \Gamma_s^{n+1} \\ \sigma(u_h^{n+1}, p_h^{n+1}) \cdot n = 0, & \Gamma_{out} \\ u_h^{n+1} = \Pi_h u_\infty^{n+1}, & \Gamma_{in} \end{cases} \tag{91}$$

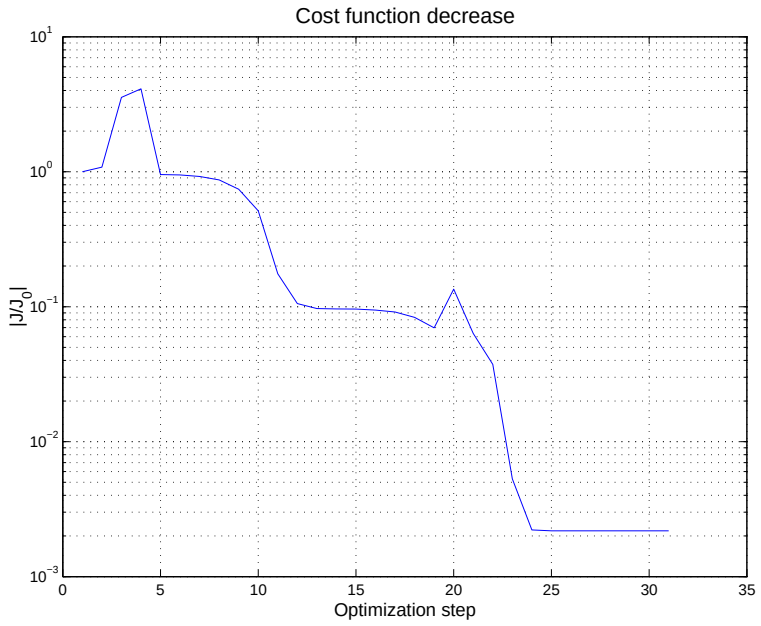

Figure 36: Cost function decrease during optimization steps

where $\chi_h^{u-w,n} = \chi_{m,h}^{u-w,n}$ stands for an approximation of the characteristic foot coming from point $x \in \Omega$ computed using the subcycling backtracking scheme,

$$\begin{cases} \chi_{i+1,h}^{u-w,n} = \chi_{i,h}^{u-w,n} - \Delta t_i \cdot (u_h^n - w_h^n)(\chi_{i,h}^{u-w,n}), & \sum_{i=1}^{m} \Delta t_i = \Delta t \\ \chi_{0,h}^{u-w,n} = x, & x \in K_h \end{cases} \tag{92}$$

We shall need the expression of the linearized system satisfied by the couple $(z,q) \stackrel{\text{def}}{=} \frac{D}{Du_\infty}(u,p) \cdot \delta u_\infty$. This can be done as in section 4. The major point, here, is that the ALE velocity w does not depend on the control variable u_∞. Hence we look for

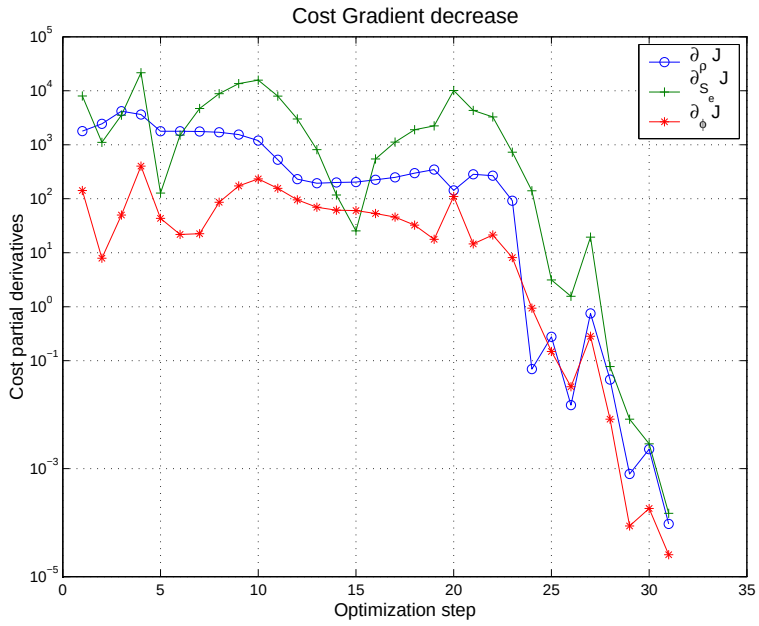

Figure 37: Gradient decrease during optimization steps

$(z_h^{n+1}, q_h^{n+1}) \in X_h^3 \times Q^h$ solution of the following system,

$$
\begin{cases}
\dfrac{1}{\Delta t}(z_h^{n+1}, v_h) + \nu\left(\nabla z_h^{n+1}, \nabla v_h\right) - (q^{n+1}, \operatorname{div} v_h) = & \\
\dfrac{1}{\Delta t}\left[\left(z_h^n(\delta g) \circ \chi_h^{u-w,n}(g), v_h\right) + \left(\nabla u_h^n(\chi_h^{u-w,n}(g)) \cdot \delta\chi_h^n, v_h\right)\right] & , \forall v_h \in V_h^{n+1} \\
\left(\operatorname{div} z_h^{n+1}, q_h\right) = 0, & \forall\, q \in Q_h^{n+1} \\
z_h^{n+1} = 0, & \Gamma_s^{n+1} \\
\sigma(z_h^{n+1}, q_h^{n+1}) \cdot n = 0, & \Gamma_{out} \\
z_h^{n+1} = \Pi_h \delta u_\infty^{n+1}, & \Gamma_{in} \\
(z_h^0, v_h) = 0, & \forall v_h \ \in V_h
\end{cases}
\tag{93}
$$

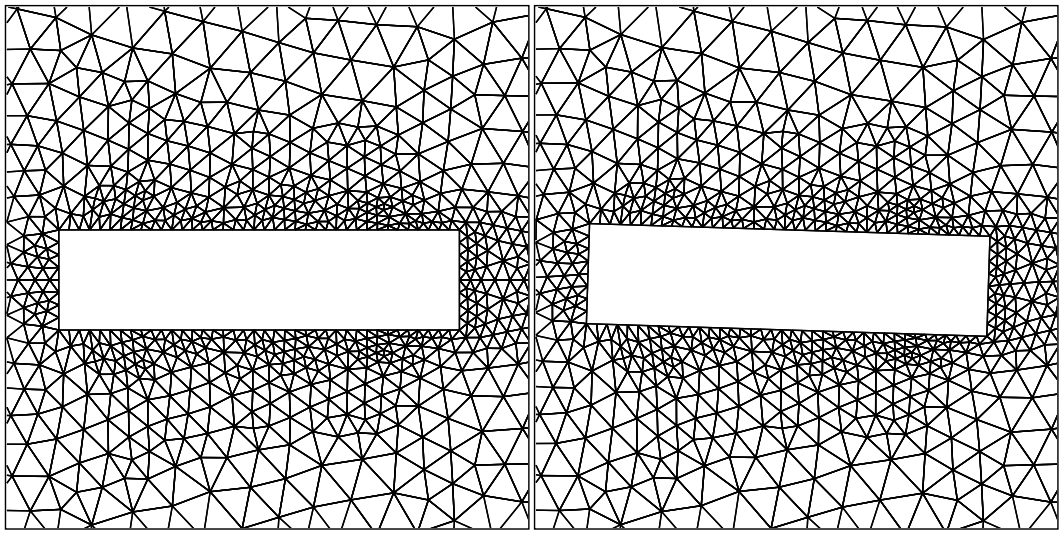

Figure 38: Mesh vertices position at t = 0s (left) and $t = 16s$ (right)

with

$$\begin{cases} \delta\chi^n_{i+1,h} = \delta\chi^n_{i,h} - \Delta t_i \cdot z^n_h(\chi^{u-w,n}_{i,h}) - \Delta t_i \nabla(u^n_h - w^n_h)(\chi^{u-w,n}_{i,h}) \cdot \delta\chi^n_{i,h}, \\ \delta\chi^n_{1,h} = -\Delta t_0 \cdot z^n_h(x), \qquad\qquad x \in \Omega \end{cases} \tag{94}$$

The derivation of the linearized system for the second order scheme follows the same arguments and has not been displayed for the sake of shortness.
These systems have been inplemented in the code LGNSI2FS and used to compute the gradient of the cost function in equation (88). The quantity,

$$D_g \stackrel{\text{def}}{=} \frac{(\nabla J^d_h - \mathrm{D}\, J^{dt}_h)}{\mathrm{D}\, J^{dt}_h}$$

has been displayed in Fig. 39 for the first and second order characteristic schemes. The results are satisfactory and can be compared with the results obtained for the driven cavity in Fig. 13.

9.4 Harmonic perturbation of the inflow velocity

As in the previous section, we use an harmonic perturbation around a steady state for the inflow velocity,

$$u_\infty(\rho, S_e, \phi) = 1 + \rho \sin(2\pi S_e t) \tag{95}$$

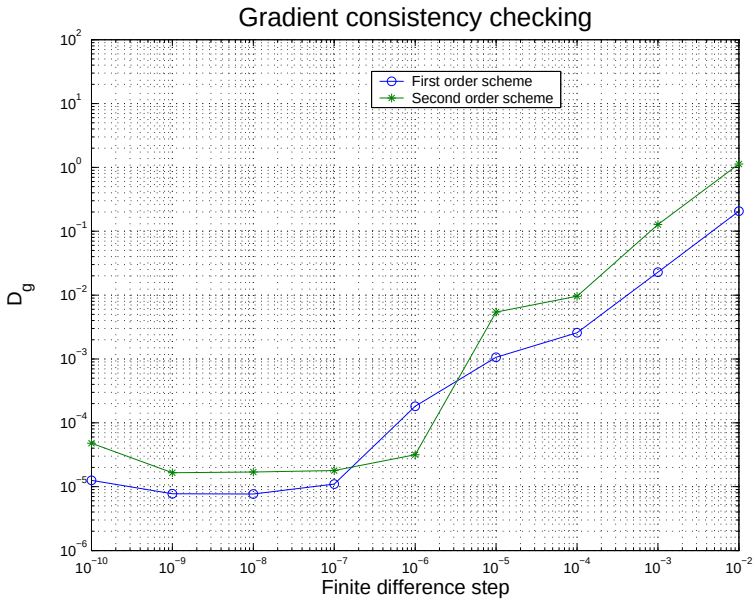

Figure 39: First and second order Lagrange-Galerkin scheme applied to the moving R4 profile - Gradient test

This means that we use a control space of dimension 2. We only deal with a synthetic load F_d produced by a set of input control parameters and try to recover this set using an optimization procedure based on a Quasi-Newton gradient based method. The moving configuration is set with $\alpha(t) = \alpha_0 \sin(2\pi\, t/T_s)$ where $\alpha_0 = 2\pi/180$ and $T_s = 5$. The synthetic load is produced with $(\rho, S_e) = (0.1, 0.1)$.
The optimization was performed using the L-BFGS-B with an initial guess in the neigbourhood of the optimal value $(\rho^0, S_e^0) = (0.15, 0.15)$. The convergence was reached after 62 iterations. The control parameters during the optimization are displayed in Fig. 40 and Fig. 41. We found this optimization problem harder than the others because the convergence only succeeded for initial guess near the target control parameters. This can be explained by the fact that this problem is stiff, since as shown in Fig.43 the value of partial derivative $\partial_{S_e} J$ at iteration 24 is greater than 10^3 while the corresponding control parameters are in a close neighbourhood of the optimal values.

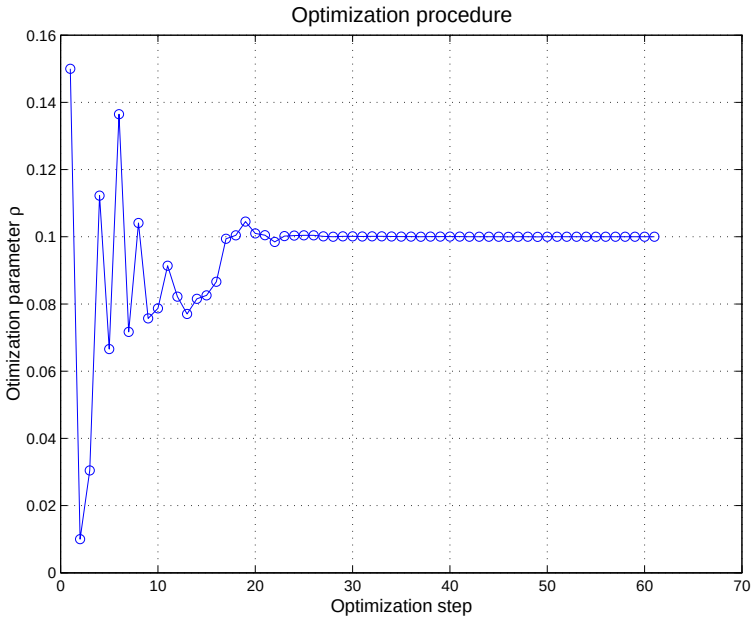

Figure 40: Amplitude control parameter during the optimization steps

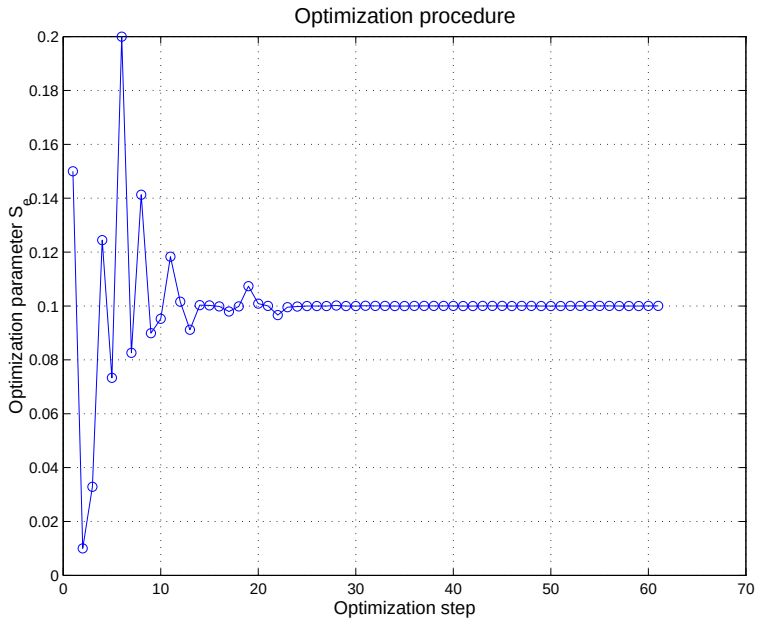

Figure 41: Frequency control parameter during the optimization steps

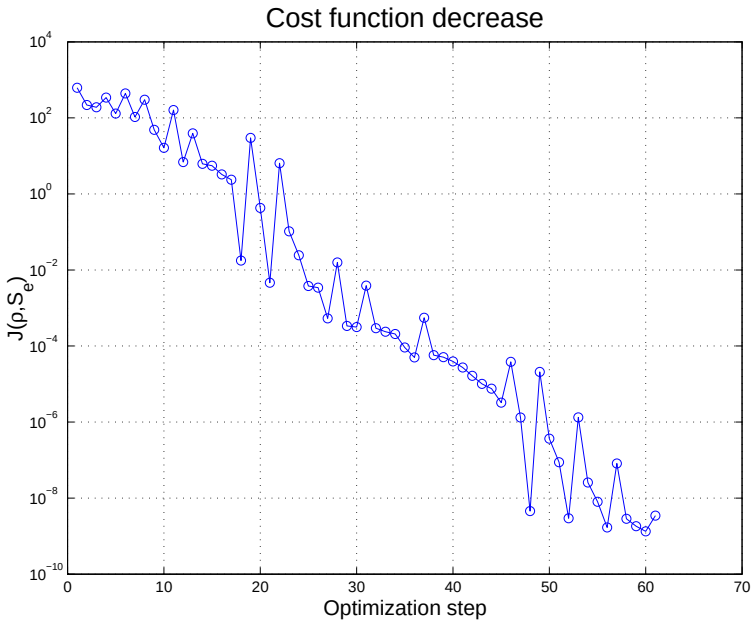

Figure 42: Cost function behaviour during the optimization steps

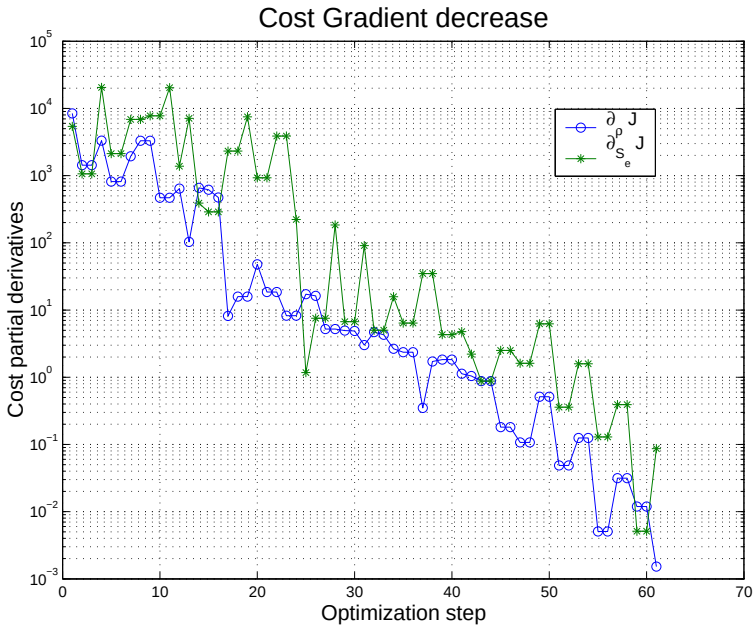

Figure 43: Gradient of the cost function behaviour during the optimization steps

10 Conclusion

In this report, we have been investigating optimal control problems for some fluid systems described by the Navier-Stokes equations in fixed and moving domains. The originality of this work is twofold :

- We have introduced a discrete linearized schemes associated to first and second order Lagrange-Galerkin discrete scheme for the Navier-Stokes system. The properties of this linearized scheme highly depend on the type of backtracking procedure used to compute the characteristic paths involved in the Lagrange-Galerkin method. We have shown that using an adaptative backtracking procedure including a projection on simplex sides along velocities leads to a non-differentiable system. This result has been numerically illustrated using the code NSI3 which is based on the adaptative backtracking procedure. Furthermore, we have shown that using a subcycling backtrackings procedure leads to differentiable problems and this has been numerically proven on simple test problems using the code LGNSI2FS.
- The computation of cost function partial derivatives has been performed using a parallel architecture taking advantage on the forward in time structure of the linearized discrete system.

The ability of our code to compute cost function gradients enabled us to deal with two optimal control problems. The numerical experiments that we performed led to the following conclusions :

- Reducing the drag around a moving circular cylinder has been feasible with the code LGNSI2FS. We have recovered early results based on different time discretization and using an adjoint based formulation. The drag minimization came with a reduction of the vortex shedding without including this objective in the cost function. The optimization worked well without any regularization, what was expected with a low Reynolds number.
- We have been also dealing with an original problem consisting in the identification of inflow boundary conditions from the knowledge of fluid loads time history on a fixed or moving embedded solid. We have shown that the exact identifiability property holds at least numerically in the case of reachable targets for the fixed and the moving configurations. For arbitrary targets, a kind of approximate identifiability property has been exhibited for the fixed configuration.

However these results have been obtained for a very small class of control parameters. This choice has been motivated by two main reasons: first for the drag minimization problem, it was proven in different works that the harmonic decomposition of boundary controls is a very good choice even with only one harmonic. Secondly, the use of a direct approach based on linearized solution computation induces an increasing number of system to solve. Even if these systems are solved in parallel, we are restricted by the number of processors that

can be used at the same time.
Again we may emphasize the fact that our aim was to validate the use of Lagrange-Galerkin algorithms while considering optimal control problems. This has been done using the direct mode approach. We are already working on the adjoint based formulations [17] which may represent a deep improvement of this work as far as efficiency is concerned.
Finally, we shall point out that the identification problem was originally designed for the fluid-structure interaction problem as described in [34]. Future works may concentrate on optimal control problems for the full coupled problems using the fluid-structure linearized system whose structure has been established in [13]. We will also deal with the adjoint formulation by using the results obtained in [35].

References

[1] F. Abergel and R. Temam. On some control problems in fluid mechanics. *Theoretical and Computational Fluid Dynamics*, 1:303–325, 1990.

[2] X. Amandolèse. *Contribution à l'étude des chargements fluides sur des obstacles non profilés fixes ou mobiles : application aux tabliers de pont.* PhD thesis, Ecole Nationale des Ponts et Chaussées, 2001.

[3] J-P. Benqué, B. Ibler, A. Keramsi, and G. Labadir. A finite element method for the Navier-Stokes equations. *Proceedings of the third international conference on finite elements in flow problems*, 1980.

[4] M. Berggren. Numerical Solution of a Flow-control Problem: Vorticity Reduction by Dynamic Boundary Action. *SIAM Journal of Scientific Computing*, 19(3):829–860, 1998.

[5] F.J Bonnans, J-C. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical optimization - Theoretical and practice aspects.* Mathématiques & Applications- Springer. xii , 1997.

[6] K. Boukir, Y. Maday, B. Métiviet, and E. Razafindrakoto. A high order characteristics / finite element method for the incompressible Navier-Stokes equations. *J. Numer. Methods Fluids, Vol. 25, pp. 1421-1454*, 1997.

[7] R.H Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

[8] M. Chevalier. *Adjoint based Control and Optimization of aerodynamic flows.* PhD thesis, Royal Institute of Technology - Department of Mechanics, 2002.

[9] M. Clerc, P. Le Tallec, and M. Mallet. Optimal Control for the Parabolized Navier-Stokes System. Technical report, INRIA, RR-2653, 1995.

[10] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Eng.*, 188(1-3):61–82, 2000.

[11] J-M. Coron. On the controllability of the 2-D incompressible Navier-Stokes equations with the Navier slip boundary conditions. *ESAIM, Control Optim. Calc. Var.*, 1:35–75, 1996.

[12] M.A Fernández. *Simplified models in fluid-structure interaction problems.* PhD thesis, University of Paris Dauphine, 2001.

[13] M.A. Fernández and M. Moubachir. Sensitivity analysis for an incompressible aeroelastic system. *Math. Models Methods Appl. Sci.*, 12(8):1109–1130, 2002.

[14] E. Fernández-Cara. On the approximate and null controllability of the Navier-Stokes equations. *SIAM Rev.*, 41(2):269–277, 1999.

[15] G. Fourestey. Stabilité des méthodes de Lagrange-Galerkin du premier et du second ordre. Technical report, INRIA, RR-4505, 2002.

[16] G. Fourestey. Une méthode des caractéristiques d'ordre deux sur maillages mobiles pour la résolution des équations de Navier-Stokes. Technical report, INRIA, RR-4448, 2002.

[17] G. Fourestey and M. Moubachir. A discrete adjoint formulation for the Navier-Stokes equations using Lagrange-Galerkin methods. *to appear*, 2003.

[18] A. Fursikov, M. Gunzburger, and L. Hou. Trace theorems for three-dimensional, time-dependent solenoidal vector fields and their applications. *Trans. Am. Math. Soc.*, 354(3):1079–1116, 2002.

[19] A.V Fursikov. *Optimal control of distributed systems. Theory and applications.* Translations of Mathematical Monographs. 187. Providence, RI: AMS, American Mathematical Society, 2000.

[20] R.H. Gallagher, G. Carey, J.T. Oden, and O.C. Zienkiewicz. *Finite Elements in Fluids Volume 6.* John Wiley - Sons, 1985.

[21] J-C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.*, 2(1):21–42, 1992.

[22] V. Girault and P.A Raviart. *Finite Element Methods for Navier-Stokes Equations.* Springer-Verlag, 1986.

[23] R. Glowinski. Finite element methods for the numerical simulation of incompressible viscous flow. Introduction to the control of the Navier-Stokes equations. In Lect. Appl. Math., editor, *Vortex dynamics and vortex methods, Proc. 21st AMS-SIAM Semin., Seattle/WA (USA)*, volume 28, pages 219–301, 1991.

[24] M.D Gunzburger and S. Manservisi. The velocity tracking problem for Navier-Stokes flows with bounded distributed controls. *SIAM Journal of Control and Optimization*, 37(6):19131–1945, 1999.

[25] M.D. Gunzburger and S. Manservisi. The velocity tracking problem for Navier-Stokes flows with boundary control. *SIAM J. Control Optimization*, 39(2):594–634, 2000.

[26] J.-W. He, R. Glowinski, R. Metcalfe, A. Nordlander, and J. Periaux. Active control and drag optimization for flow past a circular cylinder. I: Oscillatory cylinder rotation. *J. Comput. Phys.*, 163(1):83–117, 2000.

[27] J.W He and R. Glowinski. Neumann Control of Unstable Parabolic Systems: Numerical Approach. *Journal of Optimization Theory end Applications*, 96(1):1–55, 1998.

[28] J.W. He, R. Glowinski, R. Metcalfe, and J. Periaux. A numerical approach to the control and stabilization of advection-diffusion systems: Application to viscous drag reduction. *Int. J. Comput. Fluid Dyn.*, 11(1-2):131–156, 1998.

[29] C. Homescu, I.M Navon, and Z. Li. Suppresion of vortex shedding for flow around a circular cylinder using optimal control. *Int. J. Numer. Meth. Fluids.*, 38(1):43–69, 2002.

[30] O.Y Imanuvilov. Remarks on exact controllability for the Navier-Stokes equations. *ESAIM, Control Optim. Calc. Var.*, 6:39–72, 2001.

[31] D.C Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program., Ser. B*, 45(3):503–528, 1989.

[32] A. Maruoka, M. Marin, and M. Kawahara. Optimal control in Navier-Stokes equations. *International Journal of Computational Fluid Dynamics*, 9:313–322, 1998.

[33] K.W. Morton, A. Priestley, and E. Suli. Stability of the Lagrange-Galerkin Method with non-exact integration. *RAIRO Modél Math. Anal. Numér. (1988) 22, 625-653*, 1988.

[34] M. Moubachir. *Mathematical and numerical analysis of inverse and control problems for fluid-structure interaction systems.* PhD thesis, Ecole Nationale des Ponts et Chaussées, 2002.

[35] M. Moubachir and J-P. Zolésio. Optimal control of fluid-structure interaction systems : the case of a rigid body. Technical report, INRIA, to appear, 2002.

[36] J. Nocedal and S.J Wright. *Numerical optimization.* Springer Series in Operations Research, 1999.

[37] C. Pares Madronal. *Etude mathématique et approximation numérique de quelques problèmes aux limites de la mécanique des fluides.* PhD thesis, Université Paris VI, 1992.

[38] S. Piperno. *Simulation numérique de phénomènes d'interaction fluide-structure.* PhD thesis, Ecole Nationale des Ponts et Chaussées, 1995.

[39] S. Piperno. Numerical simulation of aeroelastic instabilities of elementary bridge decks. Technical report, INRIA, RR-3549, 1998.

[40] O. Pironneau. *Méthodes des éléments finis pour les fluides.* Masson, 1988.

[41] Y. Saad. *Iterative Methods for sparse Linear Systems.* http://www-users.cs.umn.edu/ saad/books.html, 2000.

[42] S.S. Sritharan, editor. *Optimal control of viscous flow.* Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. xii, 1998.

[43] C. Zhu, R.H Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, 1997.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399